```
M  4

LLL                 OOOOOOOOO        AAAAAAAAA     DDDDDDDDDDDD        SSSSSSSSSSSS     SSSSSSSSSSSS
LLL                 OOOOOOOOO        AAAAAAAAA     DDDDDDDDDDDD        SSSSSSSSSSSS     SSSSSSSSSSSS
LLL                 OOOOOOOOO        AAAAAAAAA     DDDDDDDDDDDD        SSSSSSSSSSSS     SSSSSSSSSSSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD  SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD  SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD  SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD  SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD  SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD  SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD     SSSSSSSSS        SSSSSSSSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD     SSSSSSSSS        SSSSSSSSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD     SSSSSSSSS        SSSSSSSSS
LLL              OOO         OOO  AAAAAAAAAAAAAAA  DDD         DDD              SSS              SSS
LLL              OOO         OOO  AAAAAAAAAAAAAAA  DDD         DDD              SSS              SSS
LLL              OOO         OOO  AAAAAAAAAAAAAAA  DDD         DDD              SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD              SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD              SSS              SSS
LLL              OOO         OOO  AAA         AAA  DDD         DDD              SSS              SSS
LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA  DDDDDDDDDDDD     SSSSSSSSSSSS     SSSSSSSSSSSS
LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA  DDDDDDDDDDDD     SSSSSSSSSSSS     SSSSSSSSSSSS
LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA  DDDDDDDDDDDD     SSSSSSSSSSSS     SSSSSSSSSSSS
```

_$2

```
**FILE**ID**SYSACLSRV                          I 12




  SSSSSSSS  YY      YY    SSSSSSSS    AAAAAA      CCCCCCCC  LL            SSSSSSSS  RRRRRRRR    VV      VV
  SSSSSSSS  YY      YY    SSSSSSSS    AAAAAA      CCCCCCCC  LL            SSSSSSSS  RRRRRRRR    VV      VV
SS          YY      YY  SS          AA      AA  CC          LL          SS          RR      RR  VV      VV
SS          YY      YY  SS          AA      AA  CC          LL          SS          RR      RR  VV      VV
SS            YY  YY    SS          AA      AA  CC          LL          SS          RR      RR  VV      VV
SS            YY  YY    SS          AA      AA  CC          LL          SS          RR      RR  VV      VV
  SSSSSS        YY        SSSSSS    AA      AA  CC          LL            SSSSSS    RRRRRRRR    VV      VV
  SSSSSS        YY        SSSSSS    AA      AA  CC          LL            SSSSSS    RRRRRRRR    VV      VV
        SS      YY              SS  AAAAAAAAAA  CC          LL                  SS  RR  RR      VV      VV
        SS      YY              SS  AAAAAAAAAA  CC          LL                  SS  RR  RR      VV      VV
        SS      YY              SS  AA      AA  CC          LL                  SS  RR    RR      VV  VV      ....
        SS      YY              SS  AA      AA  CC          LL                  SS  RR    RR      VV  VV      ....
SSSSSSSS        YY        SSSSSSSS  AA      AA    CCCCCCCC  LLLLLLLLLL  SSSSSSSS    RR      RR      VV          ....
SSSSSSSS        YY        SSSSSSSS  AA      AA    CCCCCCCC  LLLLLLLLLL  SSSSSSSS    RR      RR      VV          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE SYSACLSRV (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000',
0                 ADDRESSING_MODE (EXTERNAL = GENERAL,
0                                  NONEXTERNAL = LONG_RELATIVE)
0                 ) =
1 BEGIN
1
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:      Miscellaneous system services
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the routines necessary to convert an Access
1 !       Control Entry from a user (text) format to an internal (binary)
1 !       format and back again.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system, non-privileged system services.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:        L. Mark Pilant          CREATION DATE:  30-Sep-1982  11:00
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-021 LMP0299         L. Mark Pilant,         8-Aug-1984  12:31
1 !               Require SYSPRV to add an ACL to a device if it is unowned
1 !               and there is no ACL already present on the device.
1 !
1 !       V03-020 LMP0275         L. Mark Pilant,         11-Jul-1984  10:11
```

```
      0058  1 !                 If the ACL defined in the ORB is not a queue, initialize
      0059  1 !                 it.  (This is to optimize the path through EXE$CHKPRO_INT.)
      0060  1 !
      0061  1 !         V03-019 LMP0271         L. Mark Pilant,         29-Jun-1984  13:18
      0062  1 !                 Add support for ORB$V_NOACL.  (This is used by shared memory
      0063  1 !                 mailboxes to disallow ACLs.)
      0064  1 !
      0065  1 !         V03-018 LMP0261         L. Mark Pilant,         26-Jun-1984 11:25
      0066  1 !                 Fix a bug that caused the success status to be dropped from
      0067  1 !                 the formatted AUDIT type ACE.
      0068  1 !
      0069  1 !         V03-017 LMP0248         L. Mark Pilant,         3-May-1984  14:30
      0070  1 !                 Include the FID in the lock resource name for files, so that
      0071  1 !                 unique lock names will be generated.  Also, do the unlock
      0072  1 !                 regardless of the status if indicated by the item list.
      0073  1 !
      0074  1 !         V03-016 LMP0243         L. Mark Pilant,         1-May-1984  8:45
      0075  1 !                 Fix the dispatching so that if an error occurs, processing
      0076  1 !                 of the item list is not aborted.  However, only items that
      0077  1 !                 do not alter the ACL are processed.
      0078  1 !
      0079  1 !         V03-015 ACG0417         Andrew C. Goldstein,    18-Apr-1984  11:50
      0080  1 !                 Fix probing and lock access mode problems
      0081  1 !
      0082  1 !         V03-014 ACG0415         Andrew C. Goldstein,    30-Mar-1984  17:49
      0083  1 !                 Add ACL mutex handling to CHANGE_ACL; add options parsing
      0084  1 !                 to default protection ACE; break out ACL processing
      0085  1 !                 subroutines into separate common module; mask success
      0086  1 !                 and fail flags in audit display; remove IOSB parameter
      0087  1 !                 in $CHANGE_ACL; defer unlocking of file ACL until end
      0088  1 !                 of operation; find UCB ACL listhead in ORB instead of UCB.
      0089  1 !
      0090  1 !         V03-013 LMP0222         L. Mark Pilant,         27-Mar-1984  14:22
      0091  1 !                 Change the tie off symbols, necessary because of linking
      0092  1 !                 with XQP module ALLOCB, from locations to offsets.
      0093  1 !
      0094  1 !         V03-012 LMP0213         L. Mark Pilant,         13-Mar-1984  9:53
      0095  1 !                 Add support for exclusively locking and unlocking the
      0096  1 !                 object for ACL modifications.
      0097  1 !
      0098  1 !         V03-011 LMP0185         L. Mark Pilant,         2-Feb-1984  16:25
      0099  1 !                 Add support for device ACLs.  Also, improve the error
      0100  1 !                 handling considerably through judicious use of PROBing.
      0101  1 !
      0102  1 !         V03-010 LMP0179         L. Mark Pilant,         8-Dec-1983  15:46
      0103  1 !                 Add a new routine SYS$CHANGE_ACL, for changing the ACL
      0104  1 !                 associated with an object.
      0105  1 !
      0106  1 !         V03-009 LMP0174         L. Mark Pilant,         2-Dec-1983  9:58
      0107  1 !                 Add support for RMS journal-ID ACEs.
      0108  1 !
      0109  1 !         V03-008 LMP0170         L. Mark Pilant,         1-Dec-1983  16:43
      0110  1 !                 Add support for the NOPROPAGATE flag.
      0111  1 !
      0112  1 !         V03-007 LMP0152         L. Mark Pilant,         12-Sep-1983  15:13
      0113  1 !                 Make SECURITY the journal name for AUDIT and ALARM ACEs.
      0114  1 !
```

```
1 !         V03-006 LMP0140         L. Mark Pilant,         23-Aug-1983  20:21
1 !                 Add support for alphanumeric UICs.
1 !
1 !         V03-005 LMP0135         L. Mark Pilant,          8-Aug-1983  11:03
1 !                 Change the parsing and formatting of directory default
1 !                 ACEs slightly.
1 !
1 !         V03-004 LMP0123         L. Mark Pilant,         22-Jun-1983  10:36
1 !                 Change the name of the FLAGS field to OPTIONS.
1 !
1 !         V03-003 LMP0122         L. Mark Pilant,         20-Jun-1983  9:14
1 !                 Add support for a directory default protection ACE.
1 !
1 !         V03-002 LMP0114         L. Mark Pilant,         11-May-1983  10:42
1 !                 Add support for an access bitmask name table.
1 !
1 !         V03-001 LMP0103         L. Mark Pilant,         24-Apr-1983  19:14
1 !                 Add support for HIDDEN and PROTECTED ACEs.
1 !
1 !**
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1 LIBRARY 'SYS$LIBRARY:TPAMAC.L32';
```

```
1 ! Declare necessary builtin functions.
1
1 BUILTIN
1         TESTBITSC,
1         INSQUE,
1         MOVPSL,
1         MTPR,
1         PROBER,
1         PROBEW,
1         REMQUE;
1
1 LINKAGE
1         L_PROBE         = JSB (REGISTER = 3, REGISTER = 1, REGISTER = 0)
1                         : NOPRESERVE (2)
1                           NOTUSED (4, 5, 6, 7, 8, 9, 10, 11),
1
1         L_VERIFY        = JSB (REGISTER = 0; REGISTER = 1)
1                         : NOPRESERVE (2, 3)
1                           NOTUSED (4, 5, 6, 7, 8, 9, 10, 11),
1
1         L_MUTEX         = JSB (REGISTER = 0, REGISTER = 4)
1                         : NOTUSED (5, 6, 7, 8, 9, 10, 11);
1
1 FORWARD ROUTINE
1         SYS$PARSE_ACL,                          ! Convert ACE to binary
1         SYS$FORMAT_ACL,                         ! Convert ACE to text
1         SYS$CHANGE_ACL,                         ! Change an object's ACL
1         GET_PARENT_LOCK,                        ! Take out parent for ACL locks
1
1 ! TPARSE action routine
1
1         SET_ID,                                 ! Save a converted identifier
1         SET_ACCESS_BIT,                         ! Set desired access bit by name
1
1 ! ACL queue head locating routines.
1
1         GET_UCB_ACL,                            ! For UCBs
1         GET_JBC_ACL,                            ! For Job controller queue
1         GET_CEB_ACL,                            ! For CEBs
1         GET_LNT_ACL,                            ! For logical name tables
1         GET_PCB_ACL,                            ! For processes
1         GET_GBL_ACL,                            ! For global sections
1
1 ! ACL action routines.
1
1         ACL_DISPATCH,                           ! Main ACL function dispatcher
1         RUNDOWN_CHANGE_ACL;                     ! Clean up $CHANGE_ACL context
1
1 EXTERNAL ROUTINE
1         ACL_ADDENTRY,                           ! Add an ACE
1         ACL_DELENTRY,                           ! Delete an ACE
1         ACL_MODENTRY,                           ! Modify an ACE
1         ACL_FINDENTRY,                          ! Locate a specific ACE
1         ACL_FINDTYPE,                           ! Locate a specific ACE type
1         ACL_DELETEACL,                          ! Delete the entire ACL
1         ACL_READACL,                            ! Read the ACL
1         ACL_ACLLENGTH,                          ! Get the ACL's length
```

```
  1          ACL_READACE,                                    ! Read a single ACE
  1          ACL_LOCATEACE,                                  ! Locate ACE by context
  1          ACL_INIT_QUEUE,                                 ! Initialize the ACL queue
  1          ALLOC_PAGED,                                    ! Paged pool allocator
  1          DALLOC_PAGED,                                   ! Paged pool deallocator
  1          LIB$TPARSE,                                     ! General purpose parser
  1          LIB$FID_TO_NAME,                                ! FID to file-spec translator
  1          LIB$GET_VM,                                     ! General memory allocater
  1          LIB$FREE_VM,                                    ! General memory deallocater
  1          EXE$PROBER      : L_PROBE ADDRESSING_MODE (GENERAL),
  1                                                          ! Probe buffer for read
  1          EXE$PROBEW      : L_PROBE ADDRESSING_MODE (GENERAL),
  1                                                          ! Probe buffer for write
  1          IOC$VERIFYCHAN  : L_VERIFY ADDRESSING_MODE (GENERAL),
  1                                                          ! Verify channel number
  1          SCH$LOCKR       : L_MUTEX ADDRESSING_MODE (GENERAL),
  1                                                          ! Lock mutex for read
  1          SCH$LOCKW       : L_MUTEX ADDRESSING_MODE (GENERAL),
  1                                                          ! Lock mutex for write
  1          SCH$UNLOCK      : L_MUTEX ADDRESSING_MODE (GENERAL);
  1                                                          ! Unlock mutex
  1
  1  EXTERNAL
  1          CTL$GL_PCB      : REF $BBLOCK;                  ! Address of process PCB
  1
  1  MACRO
M 1          ARG_COUNT =
M 1                  BEGIN
M 1                  BUILTIN AP;
M 1                  .(.AP)<0,8>
M 1                  END
  1                  %,
  1
M 1          SET_IPL (LEVEL) =
M 1                  BEGIN
M 1                  BUILTIN MTPR;
M 1                  MTPR (%REF (LEVEL), PR$_IPL)
M 1                  END
  1                  %;
  1
  1  LITERAL
  1          ACL_TYPE        = 7,                            ! Must parallel [F11X.SRC]FCPDEF.B32
  1          MAX_ACL_SIZE    = 512;                          ! Max size of an ACL segment
  1
  1  LITERAL
  1          MIN_OBJECT_TYPE = MINU (ACL$C_FILE,
  1                                  ACL$C_DEVICE,
  1                                  ACL$C_JOBCTL_QUEUE,
  1                                  ACL$C_COMMON_EF_CLUSTER,
  1                                  ACL$C_LOGICAL_NAME_TABLE,
  1                                  ACL$C_PROCESS,
  1                                  ACL$C_GLOBAL_SECTION),
  1
  1          MAX_OBJECT_TYPE = MAXU (ACL$C_FILE,
  1                                  ACL$C_DEVICE,
  1                                  ACL$C_JOBCTL_QUEUE,
  1                                  ACL$C_COMMON_EF_CLUSTER,
                                        ACL$C_LOGICAL_NAME_TABLE,
                                        ACL$C_PROCESS,
                                        ACL$C_GLOBAL_SECTION),

        MIN_ACL_ATR      = MINU (ACL$C_ADDACLENT,
                                        ACL$C_DELACLENT,
                                        ACL$C_MODACLENT,
                                        ACL$C_FNDACLENT,
                                        ACL$C_FNDACETYP,
                                        ACL$C_DELETEACL,
                                        ACL$C_READACL,
                                        ACL$C_ACLLENGTH,
                                        ACL$C_READACE,
                                        ACL$C_RLOCK_ACL,
                                        ACL$C_WLOCK_ACL,
                                        ACL$C_UNLOCK_ACL),

        MAX_ACL_ATR      = MAXU (ACL$C_ADDACLENT,
                                        ACL$C_DELACLENT,
                                        ACL$C_MODACLENT,
                                        ACL$C_FNDACLENT,
                                        ACL$C_FNDACETYP,
                                        ACL$C_DELETEACL,
                                        ACL$C_READACL,
                                        ACL$C_ACLLENGTH,
                                        ACL$C_READACE,
                                        ACL$C_RLOCK_ACL,
                                        ACL$C_WLOCK_ACL,
                                        ACL$C_UNLOCK_ACL);

OWN
        JOURNAL_ACES     : BYTE INITIAL (0),          ! Journaling ACEs allowed
                                                      ! 0 = no support
                                                      ! 1 = support in
        ACE_BUFFER       : $BBLOCK [ATR$S_READACL],       ! Storage for binary ACE
        ACE_INDEX,                                    ! Index into ACE key area
        ACE_TYPE,                                     ! ACE type code
        ACE_RIGHTS,                                   ! ACE access rights
        UIC_FLAGS,                                    ! UIC conversion flags
        UIC_COUNT,                                    ! Number of UIC id's entered
        IDENTIFIER       : $BBLOCK [4],               ! Converted identifier
        ID_NAME          : $BBLOCK [DSC$C_S_BLN],         ! ID name descriptor
        ID_COUNT,                                     ! Number of identifiers given
        JOURNAL_NAME     : $BBLOCK [DSC$C_S_BLN],         ! Journal name descr
        ACCESS_FLAGS,                                 ! Audit access flags
        SYSTEM_PROT      : $BBLOCK [4],               ! System protection default
        OWNER_PROT       : $BBLOCK [4],               ! Owner protection default
        GROUP_PROT       : $BBLOCK [4],               ! Group protection default
        WORLD_PROT       : $BBLOCK [4],               ! World protection default
        BIT_NAME_TABLE   : REF BLOCKVECTOR [,DSC$C_S_BLN,BYTE],  ! Access bit name table addr
        CHANGE_ACMODE,                                ! Access mode for $CHANGE_ACL
        CALL_ACMODE,                                  ! Access mode of caller
        PARENT_ID,                                    ! Parent ID for ACL locks
        ACL_QUEUE_HEAD   : REF $BBLOCK,               ! Address of the ACL queue head
        ACL_POINTER      : REF $BBLOCK,               ! Address of current segment
        ACL_SPLIT,                                    ! Offset to ACE in segment
        ACE_POINTER      : REF $BBLOCK,               ! Address of current ACE
```

```
1               ACE_NUMBER,                                 ! Numeric position of ACE in ACL
1               ACL_AREA        : $BBLOCK [MAX_ACL_SIZE],           ! Temp storage for ACL segment
1               ACL_CONTEXT,                                ! Context used in $CHANGE_ACL
1               LOCK_RESNAM     : $BBLOCK [DSC$C_S_BLN],            ! Lock resouce name desc
1               RESNAM_TEXT     : $BBLOCK [31];             ! Actual resource name
1
1 ! Macro defining the subfields used within the resource name field.
1
1 MACRO
1               RSN_T_PREFIX    =  0,  0,  0,  0 %,         ! Lock name prefix
1               RSN_T_DEVNAM    =  8,  0,  0,  0 %,         ! Device name for device and
1                                                           !  file type objects
1               RSN_L_FID       = 24,  0, 32,  0 %,         ! File-id for lock
1               RSN_W_FID_NUM   = 24,  0, 16,  0 %,         ! File number
1               RSN_W_FID_SEQ   = 26,  0, 16,  0 %;         ! File sequence number
1
1 LITERAL
1               RSN_S_PREFIX    = 8,                        ! Size of lock name prefix
1               RSN_S_DEVNAM    = 16;                       ! Size of device name
1
1 ! Assumptions made about various fields used.
1
1 ! The following assumptions should track the definitions in
1 !             [RMS.SRC]RMSFILSTR.SDL module RJRDEF and
1 !             [VMSLIB.SRC]STARDEFAE.SDL module ACEDEF
1
1 $ASSUME (RJR$S_JNLID EQL 28);
1 $ASSUME ($BYTEOFFSET (RJR$T_VOLNAM) EQL 8);
1 $ASSUME ($BYTEOFFSET (RJR$T_FID) EQL 20);
1 $ASSUME ($BYTEOFFSET (RJR$Q_ID_DATE) EQL 28);
1
1 ! Define the default bit names.
1
1 BIND
1               DEFAULT_BITS    = UPLIT (
1                                               $DESCRIPTOR ('READ'),
1                                               $DESCRIPTOR ('WRITE'),
1                                               $DESCRIPTOR ('EXECUTE'),
1                                               $DESCRIPTOR ('DELETE'),
1                                               $DESCRIPTOR ('CONTROL'),
1                                               $DESCRIPTOR ('BIT_5'),
1                                               $DESCRIPTOR ('BIT_6'),
1                                               $DESCRIPTOR ('BIT_7'),
1                                               $DESCRIPTOR ('BIT_8'),
1                                               $DESCRIPTOR ('BIT_9'),
1                                               $DESCRIPTOR ('BIT_10'),
1                                               $DESCRIPTOR ('BIT_11'),
1                                               $DESCRIPTOR ('BIT_12'),
1                                               $DESCRIPTOR ('BIT_13'),
1                                               $DESCRIPTOR ('BIT_14'),
1                                               $DESCRIPTOR ('BIT_15'),
1                                               $DESCRIPTOR ('BIT_16'),
1                                               $DESCRIPTOR ('BIT_17'),
1                                               $DESCRIPTOR ('BIT_18'),
1                                               $DESCRIPTOR ('BIT_19'),
1                                               $DESCRIPTOR ('BIT_20'),
1                                               $DESCRIPTOR ('BIT_21'),
                                        $DESCRIPTOR ('BIT_22'),
                                        $DESCRIPTOR ('BIT_23'),
                                        $DESCRIPTOR ('BIT_24'),
                                        $DESCRIPTOR ('BIT_25'),
                                        $DESCRIPTOR ('BIT_26'),
                                        $DESCRIPTOR ('BIT_27'),
                                        $DESCRIPTOR ('BIT_28'),
                                        $DESCRIPTOR ('BIT_29'),
                                        $DESCRIPTOR ('BIT_30'),
                                        $DESCRIPTOR ('BIT_31')
                                        ) : VECTOR,

            LOCK_PREFIX     = UPLIT (
                                        $DESCRIPTOR ('ACL$LOCK'),
                                        $DESCRIPTOR ('ACL$FIL_'),
                                        $DESCRIPTOR ('ACL$DEV_'),
                                        $DESCRIPTOR ('ACL$JBC_'),
                                        $DESCRIPTOR ('ACL$CEF_'),
                                        $DESCRIPTOR ('ACL$LNT_'),
                                        $DESCRIPTOR ('ACL$PRC_'),
                                        $DESCRIPTOR ('ACL$GBL_')
                                        ) : VECTOR;
```

```
  1 %SBTTL  'TPARSE tables for $PARSE_ACL'
  1 ! TPARSE tables to parse an Access Control List (ACL) entry.
  1
  1 $INIT_STATE     (ACE_STATE, ACE_KEY);
  1
  1 ! Determine the type of ACE
  1
P 1 $STATE  (,
P 1         ('(')
  1         );
  1
P 1 $STATE  (GET_KEYWORD,
P 1         ('IDENTIFIER',GET_ID,,ACE$C_KEYID,ACE_TYPE),
P 1         ('BI_JOURNAL_NAME',GET_JNL,,ACE$C_BIJNL,ACE_TYPE),
P 1         ('AI_JOURNAL_NAME',GET_JNL,,ACE$C_AIJNL,ACE_TYPE),
P 1         ('AT_JOURNAL_NAME',GET_JNL,,ACE$C_ATJNL,ACE_TYPE),
P 1         ('AUDIT_JOURNAL',GET_AUDIT,,ACE$C_AUDIT,ACE_TYPE),
P 1         ('ALARM_JOURNAL',GET_ALARM,,ACE$C_ALARM,ACE_TYPE),
P 1         ('ACCESS',GET_ACCESS),
P 1         ('OPTIONS',GET_FLAGS),
P 1         ('DEFAULT_PROTECTION',GET_PROT,,ACE$C_DIRDEF,ACE_TYPE)
  1         );
P 1 $STATE  (,
P 1         (',',GET_KEYWORD),
P 1         (')',CHK_FOR_END)
  1         );
  1
  1 ! Access Control Entry.
  1
P 1 $STATE  (GET_ID,
P 1         ('='),
P 1         (':')
  1         );
P 1 $STATE  (GET_IDTYPE,
P 1         (TPA$_IDENT,,,,IDENTIFIER)
  1         );
  1
  1 ! Check for the end of the identifier.
  1
P 1 $STATE  (CHK_ENDID,
P 1         (',',GET_KEYWORD,SET_ID),
P 1         ('+',GET_IDTYPE,SET_ID),
P 1         (')',CHK_FOR_END,SET_ID)
  1         );
  1
  1 ! RMS Journal name
  1
P 1 $STATE  (GET_JNL,
P 1         ('='),
P 1         (':')
  1         );
P 1 $STATE  (,
P 1         ((GET_STRING),,,,JOURNAL_NAME)
  1         );
  1
  1 ! Check for the end of the journal name.
  1
```

```
;  447        P 0445  1 $STATE   (,
;  448        P 0446  1          (')',CHK_FOR_END)
;  449          0447  1          );
;  450          0448  1
;  451          0449  1 ! File access audit.
;  452          0450  1
;  453        P 0451  1 $STATE   (GET_AUDIT,
;  454        P 0452  1          ('='),
;  455        P 0453  1          (':')
;  456          0454  1          );
;  457        P 0455  1 $STATE   (,
;  458        P 0456  1          ((GET_STRING),,,,JOURNAL_NAME)
;  459          0457  1          );
;  460          0458  1
;  461          0459  1 ! Check to see if there is an access type to follow
;  462          0460  1
;  463        P 0461  1 $STATE   (,
;  464        P 0462  1          (',',GET_KEYWORD),
;  465        P 0463  1          (')',CHK_FOR_END)
;  466          0464  1          );
;  467          0465  1
;  468          0466  1 ! File access alarm
;  469          0467  1
;  470        P 0468  1 $STATE   (GET_ALARM,
;  471        P 0469  1          ('='),
;  472        P 0470  1          (':')
;  473          0471  1          );
;  474        P 0472  1 $STATE   (,
;  475        P 0473  1          ((GET_STRING),,,,JOURNAL_NAME)
;  476          0474  1          );
;  477          0475  1
;  478          0476  1 ! Check to see if there is an access type to follow
;  479          0477  1
;  480        P 0478  1 $STATE   (,
;  481        P 0479  1          (',',GET_KEYWORD),
;  482        P 0480  1          (')',CHK_FOR_END)
;  483          0481  1          );
;  484          0482  1
;  485          0483  1 ! Get the access type code
;  486          0484  1
;  487        P 0485  1 $STATE   (GET_ACCESS,
;  488        P 0486  1          ('='),
;  489        P 0487  1          (':')
;  490          0488  1          );
;  491        P 0489  1 $STATE   (GET_ACCTYPE,
;  492        P 0490  1          ('SUCCESS',,,ACE$M_SUCCESS,ACCESS_FLAGS),
;  493        P 0491  1          ('FAILURE',,,ACE$M_FAILURE,ACCESS_FLAGS),
;  494        P 0492  1          ('NONE'),
;  495        P 0493  1          ((GET_STRING),,SET_ACCESS_BIT)
;  496          0494  1          );
;  497        P 0495  1 $STATE   (,
;  498        P 0496  1          ('+',GET_ACCTYPE),
;  499        P 0497  1          (')',CHK_FOR_END),
;  500        P 0498  1          (',',GET_KEYWORD)
;  501          0499  1          );
;  502          0500  1
;  503          0501  1 ! Get any special flags applied to the ACE.
```

```
       1
P      1 $STATE  (GET_FLAGS,
P      1          ('='),
P      1          (':')
       1          );
P      1 $STATE  (GET_FLAGTYPE,
P      1          ('DEFAULT',,,ACE$M_DEFAULT,ACE_BUFFER[ACE$W_FLAGS]),
P      1          ('HIDDEN',,,ACE$M_HIDDEN,ACE_BUFFER[ACE$W_FLAGS]),
P      1          ('PROTECTED',,,ACE$M_PROTECTED,ACE_BUFFER[ACE$W_FLAGS]),
P      1          ('NOPROPAGATE',,,ACE$M_NOPROPAGATE,ACE_BUFFER[ACE$W_FLAGS]),
P      1          ('NONE')
       1          );
P      1 $STATE  (,
P      1          ('+',GET_FLAGTYPE),
P      1          (')',CHK_FOR_END),
P      1          (',',GET_KEYWORD)
       1          );
       1
       1 ! Get the directory default protection.
       1
P      1 $STATE  (GET_PROT,
P      1          (',')
       1          );
P      1 $STATE  (GET_PROT_CLASS,
P      1          ('SYSTEM',GET_SYS_PRO),
P      1          ('OWNER',GET_OWN_PRO),
P      1          ('GROUP',GET_GRP_PRO),
P      1          ('WORLD',GET_WOR_PRO),
P      1          (TPA$_LAMBDA,GET_KEYWORD)
       1          );
P      1 $STATE  (GET_SYS_PRO,
P      1          (':'),
P      1          ('='),
P      1          (TPA$_LAMBDA,CHK_END_PRO)
       1          );
P      1 $STATE  (GET_SYS_PRO1,
P      1          ('R',GET_SYS_PRO1,,ARM$M_READ,SYSTEM_PROT),
P      1          ('W',GET_SYS_PRO1,,ARM$M_WRITE,SYSTEM_PROT),
P      1          ('E',GET_SYS_PRO1,,ARM$M_EXECUTE,SYSTEM_PROT),
P      1          ('D',GET_SYS_PRO1,,ARM$M_DELETE,SYSTEM_PROT),
P      1          ('C',GET_SYS_PRO1,,ARM$M_CONTROL,SYSTEM_PROT),
P      1          (TPA$_LAMBDA,CHK_END_PRO)
       1          );
P      1 $STATE  (GET_OWN_PRO,
P      1          (':'),
P      1          ('='),
P      1          (TPA$_LAMBDA,CHK_END_PRO)
       1          );
P      1 $STATE  (GET_OWN_PRO1,
P      1          ('R',GET_OWN_PRO1,,ARM$M_READ,OWNER_PROT),
P      1          ('W',GET_OWN_PRO1,,ARM$M_WRITE,OWNER_PROT),
P      1          ('E',GET_OWN_PRO1,,ARM$M_EXECUTE,OWNER_PROT),
P      1          ('D',GET_OWN_PRO1,,ARM$M_DELETE,OWNER_PROT),
P      1          ('C',GET_OWN_PRO1,,ARM$M_CONTROL,OWNER_PROT),
P      1          (TPA$_LAMBDA,CHK_END_PRO)
       1          );
P      1 $STATE  (GET_GRP_PRO,
;   561      P 0559  1            (':'),
;   562      P 0560  1            ('='),
;   563      P 0561  1            (TPA$_LAMBDA,CHK_END_PRO)
;   564        0562  1            );
;   565      P 0563  1 $STATE     (GET_GRP_PRO1,
;   566      P 0564  1            ('R',GET_GRP_PRO1,,ARM$M_READ,GROUP_PROT),
;   567      P 0565  1            ('W',GET_GRP_PRO1,,ARM$M_WRITE,GROUP_PROT),
;   568      P 0566  1            ('E',GET_GRP_PRO1,,ARM$M_EXECUTE,GROUP_PROT),
;   569      P 0567  1            ('D',GET_GRP_PRO1,,ARM$M_DELETE,GROUP_PROT),
;   570      P 0568  1            ('C',GET_GRP_PRO1,,ARM$M_CONTROL,GROUP_PROT),
;   571      P 0569  1            (TPA$_LAMBDA,CHK_END_PRO)
;   572        0570  1            );
;   573      P 0571  1 $STATE     (GET_WOR_PRO,
;   574      P 0572  1            (':'),
;   575      P 0573  1            ('='),
;   576      P 0574  1            (TPA$_LAMBDA,CHK_END_PRO)
;   577        0575  1            );
;   578      P 0576  1 $STATE     (GET_WOR_PRO1,
;   579      P 0577  1            ('R',GET_WOR_PRO1,,ARM$M_READ,WORLD_PROT),
;   580      P 0578  1            ('W',GET_WOR_PRO1,,ARM$M_WRITE,WORLD_PROT),
;   581      P 0579  1            ('E',GET_WOR_PRO1,,ARM$M_EXECUTE,WORLD_PROT),
;   582      P 0580  1            ('D',GET_WOR_PRO1,,ARM$M_DELETE,WORLD_PROT),
;   583      P 0581  1            ('C',GET_WOR_PRO1,,ARM$M_CONTROL,WORLD_PROT),
;   584      P 0582  1            (TPA$_LAMBDA,CHK_END_PRO)
;   585        0583  1            );
;   586        0584  1
;   587      P 0585  1 $STATE     (CHK_END_PRO,
;   588      P 0586  1            (',',GET_PROT_CLASS),
;   589      P 0587  1            (')',CHK_FOR_END)
;   590        0588  1            );
;   591        0589  1
;   592        0590  1 ! Parse off a random string.
;   593        0591  1
;   594      P 0592  1 $STATE     (GET_STRING,
;   595      P 0593  1            (',',TPA$_FAIL),
;   596      P 0594  1            (')',TPA$_FAIL),
;   597      P 0595  1            (TPA$_EOS,TPA$_FAIL),
;   598      P 0596  1            ((GET_STRING1))
;   599        0597  1            );
;   600      P 0598  1 $STATE     (GET_STRING1,
;   601      P 0599  1            ((CHK_DELIM),GET_STRING1),
;   602      P 0600  1            (TPA$_LAMBDA,TPA$_EXIT)
;   603        0601  1            );
;   604      P 0602  1 $STATE     (CHK_DELIM,
;   605      P 0603  1            ('+',TPA$_FAIL),
;   606      P 0604  1            (',',TPA$_FAIL),
;   607      P 0605  1            (')',TPA$_FAIL),
;   608      P 0606  1            (TPA$_EOS,TPA$_FAIL),
;   609      P 0607  1            (TPA$_ANY,TPA$_EXIT)
;   610        0608  1            );
;   611        0609  1
;   612        0610  1 ! Check for the end of the ACE.  Trailing blanks are allowed.
;   613        0611  1
;   614      P 0612  1 $STATE     (CHK_FOR_END,
;   615      P 0613  1            (TPA$_EOS,TPA$_EXIT),
;   616        0614  1            );
```

```
1 %SBTTL  '$PARSE_ACL system service'
1 GLOBAL ROUTINE SYS$PARSE_ACL (ACL_STRING, ACL_ENTRY, ERROR_POSITION, BIT_TABLE) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine converts the Access Control Entry from a text form to
1 !       the binary form.
1 !
1 ! CALLING SEQUENCE:
1 !       SYS$PARSE_ACL (ARG1, ARG2, ARG3, ARG4)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of the input text descriptor
1 !       ARG2: address of the output buffer descriptor
1 !       ARG4: address of an access bit name table
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       ARG3: number of characters processed
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       SS$_NORMAL:     The conversion was successful.
1 !       SS$_ACCVIO:     The input or output descrtptors cannot be read, the
1 !                       output buffer cannot be written, or the output buffer
1 !                       is not large enough to contain the converted ACE.
1 !       SS$_IVACL:      The syntax of the input ACE is invalid.
1 !       SS$_NOSUCHID:   The identifier specified in the ACE is not in the
1 !                       rights database.
1 !
1 ! SIDE EFFECTS:
1 !       none
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       ACL_STRING      : REF $BBLOCK,          ! Address of input descriptor
2       ACL_ENTRY       : REF $BBLOCK,          ! Address of output descriptor
2       ERROR_POSITION  : REF VECTOR [,WORD],   ! Syntax error position
2       BIT_TABLE       : REF VECTOR;           ! Address of access bit ame tale
2
2 LOCAL
2       STATUS,                                 ! Routine exit status
2       ACL_STRING_LEN,                         ! Length of input ACL string
2       ACL_ENTRY_LEN,                          ! Length of output ACE buffer
2       TPARSE_BLOCK    : $BBLOCK [TPA$K_LENGTH0];      ! Parser context block
2
2 EXTERNAL LITERAL
2       LIB$_SYNTAXERR;
```

```
2
2 ! Check to see if an access bit name table was supplied.  If so, use it
2 ! rather than the default table.
2
2 BIT_NAME_TABLE = 0;
2 IF .BIT_TABLE NEQA 0
2 THEN IF PROBER (%REF (0), %REF (256), .BIT_TABLE)
2         THEN BIT_NAME_TABLE = .BIT_TABLE
2         ELSE RETURN SS$_ACCVIO;
2
2 ! Set up initial parameters.
2
2 CH$FILL (0, ATR$S_READACL, ACE_BUFFER);
2 ACE_INDEX = ACE_TYPE = ACE_RIGHTS = 0;
2 UIC_FLAGS = UIC_COUNT = 0;
2 ID_COUNT = 0;
2 ACCESS_FLAGS = 0;
2 JOURNAL_NAME[DSC$W_LENGTH] = ID_NAME[DSC$W_LENGTH] = 0;
2 SYSTEM_PROT = OWNER_PROT = GROUP_PROT = WORLD_PROT = 0;
2
2 CH$FILL (0, TPA$K_LENGTH0, TPARSE_BLOCK);
2 TPARSE_BLOCK[TPA$L_COUNT] = TPA$K_COUNT0;
2 TPARSE_BLOCK[TPA$V_ABBREV] = 1;
2 IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_STRING)
2 THEN
3     BEGIN
3     ACL_STRING_LEN = TPARSE_BLOCK[TPA$L_STRINGCNT] = .ACL_STRING[DSC$W_LENGTH];
3     IF EXE$PROBER (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
3     THEN TPARSE_BLOCK[TPA$L_STRINGPTR] = .ACL_STRING[DSC$A_POINTER]
3     ELSE RETURN SS$_ACCVIO;
3     END
2 ELSE RETURN SS$_ACCVIO;
2
2 STATUS = LIB$TPARSE (TPARSE_BLOCK, ACE_STATE, ACE_KEY);
2
2 ! If necessary set the number of characters processed.
2
2 IF .ERROR_POSITION NEQA 0
2 THEN IF PROBEW (%REF (0), %REF (2), .ERROR_POSITION)
2         THEN ERROR_POSITION[0] = .ACL_STRING_LEN -
2                                  .TPARSE_BLOCK[TPA$L_STRINGCNT]
2         ELSE RETURN SS$_ACCVIO;
2
2 ! If there 1) are any syntax errors, 2) is an invalid ACE type (zero),
2 ! or 3) is remaining text to the ACE; return an error.
2
2 IF .STATUS EQL LIB$_SYNTAXERR
2 OR .ACE_TYPE EQL 0
3 OR (.STATUS AND .TPARSE_BLOCK[TPA$L_STRINGCNT] GTR 0)
2 THEN RETURN SS$_IVACL;
2
2 IF NOT .STATUS THEN RETURN .STATUS;
2
2 ! Set up the standard ACE fields.
2
2 ACE_BUFFER[ACE$B_TYPE] = .ACE_TYPE;
2 ACE_BUFFER[ACE$W_FLAGS] = .ACE_BUFFER[ACE$W_FLAGS]
```

```
                                        OR .UIC_FLAGS
                                        OR .ACCESS_FLAGS;
ACE_BUFFER[ACE$L_ACCESS] = .ACE_RIGHTS;

! Based upon the type code, finish up the ACE.  Then do the final error
! checking to make sure that I didn't get more than I wanted.

CASE .ACE_TYPE FROM ACE$C_KEYID TO ACE$C_DIRDEF OF
SET
    [ACE$C_KEYID]:
        BEGIN
        IF .ACCESS_FLAGS NEQ 0
        OR .JOURNAL_NAME[DSC$W_LENGTH] NEQ 0
        OR .UIC_COUNT GTR 1
        OR .ACE_INDEX EQL 0
        THEN RETURN SS$_IVACL;
        ACE_BUFFER[ACE$B_SIZE] = ACE$C_LENGTH + .ACE_INDEX * 4;
        END;
    [ACE$C_BIJNL,
     ACE$C_AIJNL,
     ACE$C_ATJNL]:
        BEGIN
        IF NOT .JOURNAL_ACES THEN RETURN SS$_IVACL;
        IF .UIC_COUNT NEQ 0
        OR .ID_COUNT NEQ 0
        OR .ACCESS_FLAGS NEQ 0
        OR .ACE_RIGHTS NEQ 0
        THEN RETURN SS$_IVACL;
        CH$MOVE (.JOURNAL_NAME[DSC$W_LENGTH],
                 .JOURNAL_NAME[DSC$A_POINTER],
                 ACE_BUFFER[ACE$L_ACCESS]);
        ACE_BUFFER[ACE$B_SIZE] = .JOURNAL_NAME[DSC$W_LENGTH] +
                                 $BYTEOFFSET (ACE$L_ACCESS);
        END;
    [ACE$C_AUDIT,
     ACE$C_ALARM]:
        BEGIN
        IF .UIC_COUNT NEQ 0
        OR .ID_COUNT NEQ 0
        OR .JOURNAL_NAME[DSC$W_LENGTH] NEQ %CHARCOUNT ('SECURITY')
        OR CH$NEQ (%CHARCOUNT ('SECURITY'), UPLIT ('SECURITY'),
                   .JOURNAL_NAME[DSC$W_LENGTH], .JOURNAL_NAME[DSC$A_POINTER], 0)
        THEN RETURN SS$_IVACL;
        CH$MOVE (.JOURNAL_NAME[DSC$W_LENGTH],
                 .JOURNAL_NAME[DSC$A_POINTER],
                 ACE_BUFFER[ACE$L_KEY]);
        ACE_BUFFER[ACE$B_SIZE] = ACE$C_LENGTH + .JOURNAL_NAME[DSC$W_LENGTH];
        END;
    [ACE$C_DIRDEF]:
        BEGIN
        IF .ACCESS_FLAGS NEQ 0
        OR .JOURNAL_NAME[DSC$W_LENGTH] NEQ 0
        OR .UIC_COUNT NEQ 0
        OR .ID_COUNT NEQ 0
        THEN RETURN SS$_IVACL;
        SYSTEM_PROT = NOT .SYSTEM_PROT;
        SYSTEM_PROT[ARM$V_FILL] = 0;
```

```
3               ACE_BUFFER[ACE$L_SYS_PROT] = .SYSTEM_PROT;
3               OWNER_PROT = NOT .OWNER_PROT;
3               OWNER_PROT[ARM$V_FILL] = 0;
3               ACE_BUFFER[ACE$L_OWN_PROT] = .OWNER_PROT;
3               GROUP_PROT = NOT .GROUP_PROT;
3               GROUP_PROT[ARM$V_FILL] = 0;
3               ACE_BUFFER[ACE$L_GRP_PROT] = .GROUP_PROT;
3               WORLD_PROT = NOT .WORLD_PROT;
3               WORLD_PROT[ARM$V_FILL] = 0;
3               ACE_BUFFER[ACE$L_WOR_PROT] = .WORLD_PROT;
3               ACE_BUFFER[ACE$B_SIZE] = ACE$C_LENGTH + 16;
2               END;
2           [INRANGE,
2            OUTRANGE]: RETURN SS$_IVACL;
2   TES;
2
2   ! Check to make sure there is room to receive the ACE.
2
2   IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_ENTRY)
2   THEN
3       BEGIN
3       ACL_ENTRY_LEN = .ACL_ENTRY[DSC$W_LENGTH];
3       IF .ACE_BUFFER[ACE$B_SIZE] LEQU .ACL_ENTRY_LEN
3       AND EXE$PROBEW (0, .ACL_ENTRY_LEN, .ACL_ENTRY[DSC$A_POINTER])
3       THEN CH$COPY (.ACE_BUFFER[ACE$B_SIZE], ACE_BUFFER, 0,
3                     .ACL_ENTRY[DSC$W_LENGTH], .ACL_ENTRY[DSC$A_POINTER])
3       ELSE RETURN SS$_ACCVIO;
3       END
2   ELSE RETURN SS$_ACCVIO;
2
2   RETURN SS$_NORMAL;
2
1   END;                                                    ! End of routine SYS$PARSE_ACL

                                                              .TITLE  SYSACLSRV
                                                              .IDENT  \V04-000\

                                                              .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                                    00000 ;TPA$KEYST0
                                                          U.3:    .BLKB   0
                52  45  49  46  49  54  4E  45  44  49  00000 ;TPA$KEYST
                                                          U.5:    .ASCII  \IDENTIFIER\                      ;
                                                FF  0000A         .BYTE   -1                                ;
                                                    0000B ;TPA$KEYST0
                                                          U.11:   .BLKB   0
45  4D  41  4E  5F  4C  41  4E  52  55  4F  4A  5F  49  42  0000B ;TPA$KEYST
                                                          U.13:   .ASCII  \BI_JOURNAL_NAME\                 ;
                                                FF  0001A         .BYTE   -1                                ;
                                                    0001B ;TPA$KEYST0
                                                          U.19:   .BLKB   0
45  4D  41  4E  5F  4C  41  4E  52  55  4F  4A  5F  49  41  0001B ;TPA$KEYST
                                                          U.21:   .ASCII  \AI_JOURNAL_NAME\                 ;
                                                FF  0002A         .BYTE   -1                                ;
                                                    0002B ;TPA$KEYST0
                                                          U.26:   .BLKB   0
```

```
45 4D 41 4E 5F 4C 41 4E 52 55 4F 4A 5F 54 41  0002B ;TPA$KEYST
                                                     U.28:    .ASCII   \AT_JOURNAL_NAME\                          ;
                                           FF  0003A          .BYTE    -1                                         ;
                                               0003B ;TPA$KEYST0
                                                     U.33:    .BLKB    0
   4C 41 4E 52 55 4F 4A 5F 54 49 44 55 41  0003B ;TPA$KEYST
                                                     U.35:    .ASCII   \AUDIT_JOURNAL\                            ;
                                           FF  00048          .BYTE    -1                                         ;
                                               00049 ;TPA$KEYST0
                                                     U.41:    .BLKB    0
   4C 41 4E 52 55 4F 4A 5F 4D 52 41 4C 41  00049 ;TPA$KEYST
                                                     U.43:    .ASCII   \ALARM_JOURNAL\                            ;
                                           FF  00056          .BYTE    -1                                         ;
                                               00057 ;TPA$KEYST0
                                                     U.49:    .BLKB    0
                           53 53 45 43 43 41  00057 ;TPA$KEYST
                                                     U.51:    .ASCII   \ACCESS\                                   ;
                                           FF  0005D          .BYTE    -1                                         ;
                                               0005E ;TPA$KEYST0
                                                     U.55:    .BLKB    0
                        53 4E 4F 49 54 50 4F  0005E ;TPA$KEYST
                                                     U.57:    .ASCII   \OPTIONS\                                  ;
                                           FF  00065          .BYTE    -1                                         ;
                                               00066 ;TPA$KEYST0
                                                     U.61:    .BLKB    0
54 43 45 54 4F 52 50 5F 54 4C 55 41 46 45 44  00066 ;TPA$KEYST
                                                     U.63:    .ASCII   \DEFAULT_PROTECTION\                       ;
                                     4E 4F 49  00075                                                              ;
                                           FF  00078          .BYTE    -1                                         ;
                                           FF  00079 ;TPA$KEYFILL
                                                     U.69:    .BYTE    -1                                         ;
                                               0007A ;TPA$KEYST0
                                                     U.116:   .BLKB    0
                        53 53 45 43 43 55 53  0007A ;TPA$KEYST
                                                     U.118:   .ASCII   \SUCCESS\                                  ;
                                           FF  00081          .BYTE    -1                                         ;
                                               00082 ;TPA$KEYST0
                                                     U.122:   .BLKB    0
                        45 52 55 4C 49 41 46  00082 ;TPA$KEYST
                                                     U.124:   .ASCII   \FAILURE\                                  ;
                                           FF  00089          .BYTE    -1                                         ;
                                               0008A ;TPA$KEYST0
                                                     U.128:   .BLKB    0
                                  45 4E 4F 4E  0008A ;TPA$KEYST
                                                     U.130:   .ASCII   \NONE\                                     ;
                                           FF  0008E          .BYTE    -1                                         ;
                                           FF  0008F ;TPA$KEYFILL
                                                     U.135:   .BYTE    -1                                         ;
                                               00090 ;TPA$KEYST0
                                                     U.144:   .BLKB    0
                        54 4C 55 41 46 45 44  00090 ;TPA$KEYST
                                                     U.146:   .ASCII   \DEFAULT\                                  ;
                                           FF  00097          .BYTE    -1                                         ;
                                               00098 ;TPA$KEYST0
                                                     U.150:   .BLKB    0
                           4E 45 44 44 49 48  00098 ;TPA$KEYST
                                                     U.152:   .ASCII   \HIDDEN\                                   ;
                                     FF  0009E        .BYTE    -1                                          ;
                                         0009F ;TPA$KEYST0
                                               U.156:  .BLKB    0
          44  45  54  43  45  54  4F  52  50  0009F ;TPA$KEYST
                                               U.158:  .ASCII   \PROTECTED\                                ;
                                     FF  000A8        .BYTE    -1                                          ;
                                         000A9 ;TPA$KEYST0
                                               U.162:  .BLKB    0
  45  54  41  47  41  50  4F  52  50  4F  4E  000A9 ;TPA$KEYST
                                               U.164:  .ASCII   \NOPROPAGATE\                              ;
                                     FF  000B4        .BYTE    -1                                          ;
                                         000B5 ;TPA$KEYST0
                                               U.168:  .BLKB    0
                          45  4E  4F  4E  000B5 ;TPA$KEYST
                                               U.170:  .ASCII   \NONE\                                     ;
                                     FF  000B9        .BYTE    -1                                          ;
                                     FF  000BA ;TPA$KEYFILL
                                               U.172:  .BYTE    -1                                         ;
                                         000BB ;TPA$KEYST0
                                               U.180:  .BLKB    0
                  4D  45  54  53  59  53  000BB ;TPA$KEYST
                                               U.182:  .ASCII   \SYSTEM\                                   ;
                                     FF  000C1        .BYTE    -1                                          ;
                                         000C2 ;TPA$KEYST0
                                               U.186:  .BLKB    0
                      52  45  4E  57  4F  000C2 ;TPA$KEYST
                                               U.188:  .ASCII   \OWNER\                                    ;
                                     FF  000C7        .BYTE    -1                                          ;
                                         000C8 ;TPA$KEYST0
                                               U.192:  .BLKB    0
                      50  55  4F  52  47  000C8 ;TPA$KEYST
                                               U.194:  .ASCII   \GROUP\                                    ;
                                     FF  000CD        .BYTE    -1                                          ;
                                         000CE ;TPA$KEYST0
                                               U.198:  .BLKB    0
                      44  4C  52  4F  57  000CE ;TPA$KEYST
                                               U.200:  .ASCII   \WORLD\                                    ;
                                     FF  000D3        .BYTE    -1                                          ;
                                     FF  000D4 ;TPA$KEYFILL
                                               U.206:  .BYTE    -1                                         ;

                                                       .PSECT   _LIB$STATE$,NOWRT,  SHR,  PIC,1

                                         00000 ACE_STATE::
                                                       .BLKB    0
                                   0428  00000 ;TPA$TYPE
                                               U.2:    .WORD    1064                                       ;
                                         00002 GET_KEYWORD:
                                                       .BLKB    0
                                   7100  00002 ;TPA$TYPE
                                               U.6:    .WORD    28928                                      ;
                              00000000*  00004 ;TPA$ADDR
                                               U.7:    .LONG    <<ACE_TYPE-U.7>-4>                         ;
                              00000001   00008 ;TPA$MASK
                                               U.8:    .LONG    1                                          ;
                                  0000*  0000C ;TPA$TARGET
                                               U.10:   .WORD    <<U.9-U.10>-2>                             ;
                  7101  0000E  ;TPA$TYPE
                               U.14:     .WORD    28929                                   ;
             00000000*  00010  ;TPA$ADDR
                               U.15:     .LONG    <<ACE_TYPE-U.15>-4>                     ;
             00000002   00014  ;TPA$MASK
                               U.16:     .LONG    2                                       ;
                 0000*  00018  ;TPA$TARGET
                               U.18:     .WORD    <<U.17-U.18>-2>                         ;
                  7102  0001A  ;TPA$TYPE
                               U.22:     .WORD    28930                                   ;
             00000000*  0001C  ;TPA$ADDR
                               U.23:     .LONG    <<ACE_TYPE-U.23>-4>                     ;
             00000003   00020  ;TPA$MASK
                               U.24:     .LONG    3                                       ;
                 0000*  00024  ;TPA$TARGET
                               U.25:     .WORD    <<U.17-U.25>-2>                         ;
                  7103  00026  ;TPA$TYPE
                               U.29:     .WORD    28931                                   ;
             00000000*  00028  ;TPA$ADDR
                               U.30:     .LONG    <<ACE_TYPE-U.30>-4>                     ;
             00000004   0002C  ;TPA$MASK
                               U.31:     .LONG    4                                       ;
                 0000*  00030  ;TPA$TARGET
                               U.32:     .WORD    <<U.17-U.32>-2>                         ;
                  7104  00032  ;TPA$TYPE
                               U.36:     .WORD    28932                                   ;
             00000000*  00034  ;TPA$ADDR
                               U.37:     .LONG    <<ACE_TYPE-U.37>-4>                     ;
             00000005   00038  ;TPA$MASK
                               U.38:     .LONG    5                                       ;
                 0000*  0003C  ;TPA$TARGET
                               U.40:     .WORD    <<U.39-U.40>-2>                         ;
                  7105  0003E  ;TPA$TYPE
                               U.44:     .WORD    28933                                   ;
             00000000*  00040  ;TPA$ADDR
                               U.45:     .LONG    <<ACE_TYPE-U.45>-4>                     ;
             00000006   00044  ;TPA$MASK
                               U.46:     .LONG    6                                       ;
                 0000*  00048  ;TPA$TARGET
                               U.48:     .WORD    <<U.47-U.48>-2>                         ;
                  1106  0004A  ;TPA$TYPE
                               U.52:     .WORD    4358                                    ;
                 0000*  0004C  ;TPA$TARGET
                               U.54:     .WORD    <<U.53-U.54>-2>                         ;
                  1107  0004E  ;TPA$TYPE
                               U.58:     .WORD    4359                                    ;
                 0000*  00050  ;TPA$TARGET
                               U.60:     .WORD    <<U.59-U.60>-2>                         ;
                  7508  00052  ;TPA$TYPE
                               U.64:     .WORD    29960                                   ;
             00000000*  00054  ;TPA$ADDR
                               U.65:     .LONG    <<ACE_TYPE-U.65>-4>                     ;
             00000009   00058  ;TPA$MASK
                               U.66:     .LONG    9                                       ;
                 0000*  0005C  ;TPA$TARGET
                               U.68:     .WORD    <<U.67-U.68>-2>                         ;
                  102C  0005E  ;TPA$TYPE
```

```
                    U.70:     .WORD    4140                                ;
     0000*  00060  ;TPA$TARGET
                    U.71:     .WORD    <<GET_KEYWORD-U.71>-2>              ;
     1429   00062  ;TPA$TYPE
                    U.72:     .WORD    5161                                ;
     0000*  00064  ;TPA$TARGET
                    U.74:     .WORD    <<U.73-U.74>-2>                     ;
            00066  ;GET_ID
                    U.9:      .BLKB    0
     003D   00066  ;TPA$TYPE
                    U.75:     .WORD    61                                  ;
     043A   00068  ;TPA$TYPE
                    U.76:     .WORD    1082                                ;
            0006A  GET_IDTYPE:
                              .BLKB    0
     45EC   0006A  ;TPA$TYPE
                    U.77:     .WORD    17900                               ;
 00000000*  0006C  ;TPA$ADDR
                    U.78:     .LONG    <<IDENTIFIER-U.78>-4>               ;
            00070  CHK_ENDID:
                              .BLKB    0
     902C   00070  ;TPA$TYPE
                    U.79:     .WORD    -28628                              ;
 00000000V  00072  ;TPA$ACTION
                    U.80:     .LONG    <<SET_ID-U.80>-4>                   ;
     0000*  00076  ;TPA$TARGET
                    U.81:     .WORD    <<GET_KEYWORD-U.81>-2>              ;
     902B   00078  ;TPA$TYPE
                    U.82:     .WORD    -28629                              ;
 00000000V  0007A  ;TPA$ACTION
                    U.83:     .LONG    <<SET_ID-U.83>-4>                   ;
     0000*  0007E  ;TPA$TARGET
                    U.84:     .WORD    <<GET_IDTYPE-U.84>-2>               ;
     9429   00080  ;TPA$TYPE
                    U.85:     .WORD    -27607                              ;
 00000000V  00082  ;TPA$ACTION
                    U.86:     .LONG    <<SET_ID-U.86>-4>                   ;
     0000*  00086  ;TPA$TARGET
                    U.87:     .WORD    <<U.73-U.87>-2>                     ;
            00088  ;GET_JNL
                    U.17:     .BLKB    0
     003D   00088  ;TPA$TYPE
                    U.88:     .WORD    61                                  ;
     043A   0008A  ;TPA$TYPE
                    U.89:     .WORD    1082                                ;
     4DF8   0008C  ;TPA$TYPE
                    U.90:     .WORD    19960                               ;
     0000*  0008E  ;TPA$SUBEXP
                    U.92:     .WORD    <<U.91-U.92>-2>                     ;
 00000000*  00090  ;TPA$ADDR
                    U.93:     .LONG    <<JOURNAL_NAME-U.93>-4>             ;
     1429   00094  ;TPA$TYPE
                    U.94:     .WORD    5161                                ;
     0000*  00096  ;TPA$TARGET
                    U.95:     .WORD    <<U.73-U.95>-2>                     ;
            00098  ;GET_AUDIT
                    U.39:     .BLKB    0
```

```
             003D  00098 ;TPA$TYPE
                         U.96:    .WORD    61                                          ;
             043A  0009A ;TPA$TYPE
                         U.97:    .WORD    1082                                        ;
             4DF8  0009C ;TPA$TYPE
                         U.98:    .WORD    19960                                       ;
            0000*  0009E ;TPA$SUBEXP
                         U.99:    .WORD    <<U.91-U.99>-2>                             ;
        00000000*  000A0 ;TPA$ADDR
                         U.100:   .LONG    <<JOURNAL_NAME-U.100>-4>                    ;
             102C  000A4 ;TPA$TYPE
                         U.101:   .WORD    4140                                        ;
            0000*  000A6 ;TPA$TARGET
                         U.102:   .WORD    <<GET_KEYWORD-U.102>-2>                     ;
             1429  000A8 ;TPA$TYPE
                         U.103:   .WORD    5161                                        ;
            0000*  000AA ;TPA$TARGET
                         U.104:   .WORD    <<U.73-U.104>-2>                            ;
                   000AC ;GET_ALARM
                         U.47:    .BLKB    0
             003D  000AC ;TPA$TYPE
                         U.105:   .WORD    61                                          ;
             043A  000AE ;TPA$TYPE
                         U.106:   .WORD    1082                                        ;
             4DF8  000B0 ;TPA$TYPE
                         U.107:   .WORD    19960                                       ;
            0000*  000B2 ;TPA$SUBEXP
                         U.108:   .WORD    <<U.91-U.108>-2>                            ;
        00000000*  000B4 ;TPA$ADDR
                         U.109:   .LONG    <<JOURNAL_NAME-U.109>-4>                    ;
             102C  000B8 ;TPA$TYPE
                         U.110:   .WORD    4140                                        ;
            0000*  000BA ;TPA$TARGET
                         U.111:   .WORD    <<GET_KEYWORD-U.111>-2>                     ;
             1429  000BC ;TPA$TYPE
                         U.112:   .WORD    5161                                        ;
            0000*  000BE ;TPA$TARGET
                         U.113:   .WORD    <<U.73-U.113>-2>                            ;
                   000C0 ;GET_ACCESS
                         U.53:    .BLKB    0
             003D  000C0 ;TPA$TYPE
                         U.114:   .WORD    61                                          ;
             043A  000C2 ;TPA$TYPE
                         U.115:   .WORD    1082                                        ;
                   000C4 GET_ACCTYPE:
                                  .BLKB    0
             6109  000C4 ;TPA$TYPE
                         U.119:   .WORD    24841                                       ;
        00000000*  000C6 ;TPA$ADDR
                         U.120:   .LONG    <<ACCESS_FLAGS-U.120>-4>                    ;
         00000001  000CA ;TPA$MASK
                         U.121:   .LONG    1                                           ;
             610A  000CE ;TPA$TYPE
                         U.125:   .WORD    24842                                       ;
        00000000*  000D0 ;TPA$ADDR
                         U.126:   .LONG    <<ACCESS_FLAGS-U.126>-4>                    ;
         00000002  000D4 ;TPA$MASK
```

```
                          U.127:   .LONG    2                                   ;
        010B   000D8  ;TPA$TYPE
                          U.131:   .WORD    267                                 ;
        8DF8   000DA  ;TPA$TYPE
                          U.132:   .WORD    -29192                              ;
       0000*   000DC  ;TPA$SUBEXP
                          U.133:   .WORD    <<U.91-U.133>-2>                    ;
   00000000V   000DE  ;TPA$ACTION
                          U.134:   .LONG    <<SET_ACCESS_BIT-U.134>-4>          ;
        102B   000E2  ;TPA$TYPE
                          U.136:   .WORD    4139                                ;
       0000*   000E4  ;TPA$TARGET
                          U.137:   .WORD    <<GET_ACCTYPE-U.137>-2>             ;
        1029   000E6  ;TPA$TYPE
                          U.138:   .WORD    4137                                ;
       0000*   000E8  ;TPA$TARGET
                          U.139:   .WORD    <<U.73-U.139>-2>                    ;
        142C   000EA  ;TPA$TYPE
                          U.140:   .WORD    5164                                ;
       0000*   000EC  ;TPA$TARGET
                          U.141:   .WORD    <<GET_KEYWORD-U.141>-2>             ;
               000EE  ;GET_FLAGS
                          U.59:    .BLKB    0
        003D   000EE  ;TPA$TYPE
                          U.142:   .WORD    61                                  ;
        043A   000F0  ;TPA$TYPE
                          U.143:   .WORD    1082                                ;
               000F2  GET_FLAGTYPE:
                                   .BLKB    0
        610C   000F2  ;TPA$TYPE
                          U.147:   .WORD    24844                               ;
   00000000*   000F4  ;TPA$ADDR
                          U.148:   .LONG    <<<ACE_BUFFER+2>-U.148>-4>          ;
    00000100   000F8  ;TPA$MASK
                          U.149:   .LONG    256                                 ;
        610D   000FC  ;TPA$TYPE
                          U.153:   .WORD    24845                               ;
   00000000*   000FE  ;TPA$ADDR
                          U.154:   .LONG    <<<ACE_BUFFER+2>-U.154>-4>          ;
    00000400   00102  ;TPA$MASK
                          U.155:   .LONG    1024                                ;
        610E   00106  ;TPA$TYPE
                          U.159:   .WORD    24846                               ;
   00000000*   00108  ;TPA$ADDR
                          U.160:   .LONG    <<<ACE_BUFFER+2>-U.160>-4>          ;
    00000200   0010C  ;TPA$MASK
                          U.161:   .LONG    512                                 ;
        610F   00110  ;TPA$TYPE
                          U.165:   .WORD    24847                               ;
   00000000*   00112  ;TPA$ADDR
                          U.166:   .LONG    <<<ACE_BUFFER+2>-U.166>-4>          ;
    00000800   00116  ;TPA$MASK
                          U.167:   .LONG    2048                                ;
        0510   0011A  ;TPA$TYPE
                          U.171:   .WORD    1296                                ;
        102B   0011C  ;TPA$TYPE
                          U.173:   .WORD    4139                                ;
```

```
         0000*  0011E ;TPA$TARGET
                      U.174:  .WORD   <<GET_FLAGTYPE-U.174>-2>                 ;
         1029   00120 ;TPA$TYPE
                      U.175:  .WORD   4137                                     ;
         0000*  00122 ;TPA$TARGET
                      U.176:  .WORD   <<U.73-U.176>-2>                         ;
         142C   00124 ;TPA$TYPE
                      U.177:  .WORD   5164                                     ;
         0000*  00126 ;TPA$TARGET
                      U.178:  .WORD   <<GET_KEYWORD-U.178>-2>                  ;
                00128 ;GET_PROT
                      U.67:   .BLKB   0
         042C   00128 ;TPA$TYPE
                      U.179:  .WORD   1068                                     ;
                0012A GET_PROT_CLASS:
                              .BLKB   0
         1111   0012A ;TPA$TYPE
                      U.183:  .WORD   4369                                     ;
         0000*  0012C ;TPA$TARGET
                      U.185:  .WORD   <<U.184-U.185>-2>                        ;
         1112   0012E ;TPA$TYPE
                      U.189:  .WORD   4370                                     ;
         0000*  00130 ;TPA$TARGET
                      U.191:  .WORD   <<U.190-U.191>-2>                        ;
         1113   00132 ;TPA$TYPE
                      U.195:  .WORD   4371                                     ;
         0000*  00134 ;TPA$TARGET
                      U.197:  .WORD   <<U.196-U.197>-2>                        ;
         1114   00136 ;TPA$TYPE
                      U.201:  .WORD   4372                                     ;
         0000*  00138 ;TPA$TARGET
                      U.203:  .WORD   <<U.202-U.203>-2>                        ;
         15F6   0013A ;TPA$TYPE
                      U.204:  .WORD   5622                                     ;
         0000*  0013C ;TPA$TARGET
                      U.205:  .WORD   <<GET_KEYWORD-U.205>-2>                  ;
                0013E ;GET_SYS_PRO
                      U.184:  .BLKB   0
         003A   0013E ;TPA$TYPE
                      U.207:  .WORD   58                                       ;
         003D   00140 ;TPA$TYPE
                      U.208:  .WORD   61                                       ;
         15F6   00142 ;TPA$TYPE
                      U.209:  .WORD   5622                                     ;
         0000*  00144 ;TPA$TARGET
                      U.211:  .WORD   <<U.210-U.211>-2>                        ;
                00146 GET_SYS_PRO1:
                              .BLKB   0
         7052   00146 ;TPA$TYPE
                      U.212:  .WORD   28754                                    ;
     00000000*  00148 ;TPA$ADDR
                      U.213:  .LONG   <<SYSTEM_PROT-U.213>-4>                  ;
     00000001   0014C ;TPA$MASK
                      U.214:  .LONG   1                                        ;
         0000*  00150 ;TPA$TARGET
                      U.215:  .WORD   <<GET_SYS_PRO1-U.215>-2>                 ;
         7057   00152 ;TPA$TYPE
```

```
                         U.216:  .WORD   28759                                   ;
00000000* 00154 ;TPA$ADDR
                         U.217:  .LONG   <<SYSTEM_PROT-U.217>-4>                 ;
00000002  00158 ;TPA$MASK
                         U.218:  .LONG   2                                       ;
    0000* 0015C ;TPA$TARGET
                         U.219:  .WORD   <<GET_SYS_PRO1-U.219>-2>                ;
     7045 0015E ;TPA$TYPE
                         U.220:  .WORD   28741                                   ;
00000000* 00160 ;TPA$ADDR
                         U.221:  .LONG   <<SYSTEM_PROT-U.221>-4>                 ;
00000004  00164 ;TPA$MASK
                         U.222:  .LONG   4                                       ;
    0000* 00168 ;TPA$TARGET
                         U.223:  .WORD   <<GET_SYS_PRO1-U.223>-2>                ;
     7044 0016A ;TPA$TYPE
                         U.224:  .WORD   28740                                   ;
00000000* 0016C ;TPA$ADDR
                         U.225:  .LONG   <<SYSTEM_PROT-U.225>-4>                 ;
00000008  00170 ;TPA$MASK
                         U.226:  .LONG   8                                       ;
    0000* 00174 ;TPA$TARGET
                         U.227:  .WORD   <<GET_SYS_PRO1-U.227>-2>                ;
     7043 00176 ;TPA$TYPE
                         U.228:  .WORD   28739                                   ;
00000000* 00178 ;TPA$ADDR
                         U.229:  .LONG   <<SYSTEM_PROT-U.229>-4>                 ;
00000010  0017C ;TPA$MASK
                         U.230:  .LONG   16                                      ;
    0000* 00180 ;TPA$TARGET
                         U.231:  .WORD   <<GET_SYS_PRO1-U.231>-2>                ;
     15F6 00182 ;TPA$TYPE
                         U.232:  .WORD   5622                                    ;
    0000* 00184 ;TPA$TARGET
                         U.233:  .WORD   <<U.210-U.233>-2>                       ;
          00186 ;GET_OWN_PRO
                         U.190:  .BLKB   0
     003A 00186 ;TPA$TYPE
                         U.234:  .WORD   58                                      ;
     003D 00188 ;TPA$TYPE
                         U.235:  .WORD   61                                      ;
     15F6 0018A ;TPA$TYPE
                         U.236:  .WORD   5622                                    ;
    0000* 0018C ;TPA$TARGET
                         U.237:  .WORD   <<U.210-U.237>-2>                       ;
          0018E GET_OWN_PRO1:
                                 .BLKB   0
     7052 0018E ;TPA$TYPE
                         U.238:  .WORD   28754                                   ;
00000000* 00190 ;TPA$ADDR
                         U.239:  .LONG   <<OWNER_PROT-U.239>-4>                  ;
00000001  00194 ;TPA$MASK
                         U.240:  .LONG   1                                       ;
    0000* 00198 ;TPA$TARGET
                         U.241:  .WORD   <<GET_OWN_PRO1-U.241>-2>                ;
     7057 0019A ;TPA$TYPE
                         U.242:  .WORD   28759                                   ;
```

```
                                        00000000* 0019C ;TPA$ADDR
                                                        U.243:  .LONG   <<OWNER_PROT-U.243>-4>            ;
                                        00000002  001A0 ;TPA$MASK
                                                        U.244:  .LONG   2                                 ;
                                            0000* 001A4 ;TPA$TARGET
                                                        U.245:  .WORD   <<GET_OWN_PRO1-U.245>-2>          ;
                                             7045 001A6 ;TPA$TYPE
                                                        U.246:  .WORD   28741                             ;
                                        00000000* 001A8 ;TPA$ADDR
                                                        U.247:  .LONG   <<OWNER_PROT-U.247>-4>            ;
                                        00000004  001AC ;TPA$MASK
                                                        U.248:  .LONG   4                                 ;
                                            0000* 001B0 ;TPA$TARGET
                                                        U.249:  .WORD   <<GET_OWN_PRO1-U.249>-2>          ;
                                             7044 001B2 ;TPA$TYPE
                                                        U.250:  .WORD   28740                             ;
                                        00000000* 001B4 ;TPA$ADDR
                                                        U.251:  .LONG   <<OWNER_PROT-U.251>-4>            ;
                                        00000008  001B8 ;TPA$MASK
                                                        U.252:  .LONG   8                                 ;
                                            0000* 001BC ;TPA$TARGET
                                                        U.253:  .WORD   <<GET_OWN_PRO1-U.253>-2>          ;
                                             7043 001BE ;TPA$TYPE
                                                        U.254:  .WORD   28739                             ;
                                        00000000* 001C0 ;TPA$ADDR
                                                        U.255:  .LONG   <<OWNER_PROT-U.255>-4>            ;
                                        00000010  001C4 ;TPA$MASK
                                                        U.256:  .LONG   16                                ;
                                            0000* 001C8 ;TPA$TARGET
                                                        U.257:  .WORD   <<GET_OWN_PRO1-U.257>-2>          ;
                                             15F6 001CA ;TPA$TYPE
                                                        U.258:  .WORD   5622                              ;
                                            0000* 001CC ;TPA$TARGET
                                                        U.259:  .WORD   <<U.210-U.259>-2>                 ;
                                                  001CE ;GET_GRP_PRO
                                                        U.196:  .BLKB   0
                                             003A 001CE ;TPA$TYPE
                                                        U.260:  .WORD   58                                ;
                                             003D 001D0 ;TPA$TYPE
                                                        U.261:  .WORD   61                                ;
                                             15F6 001D2 ;TPA$TYPE
                                                        U.262:  .WORD   5622                              ;
                                            0000* 001D4 ;TPA$TARGET
                                                        U.263:  .WORD   <<U.210-U.263>-2>                 ;
                                                  001D6 GET_GRP_PRO1:
                                                                .BLKB   0
                                             7052 001D6 ;TPA$TYPE
                                                        U.264:  .WORD   28754                             ;
                                        00000000* 001D8 ;TPA$ADDR
                                                        U.265:  .LONG   <<GROUP_PROT-U.265>-4>            ;
                                        00000001  001DC ;TPA$MASK
                                                        U.266:  .LONG   1                                 ;
                                            0000* 001E0 ;TPA$TARGET
                                                        U.267:  .WORD   <<GET_GRP_PRO1-U.267>-2>          ;
                                             7057 001E2 ;TPA$TYPE
                                                        U.268:  .WORD   28759                             ;
                                        00000000* 001E4 ;TPA$ADDR
```

```
                               U.269:   .LONG   <<GROUP_PROT-U.269>-4>          ;
          00000002  001E8 ;TPA$MASK
                               U.270:   .LONG   2                               ;
             0000*  001EC ;TPA$TARGET
                               U.271:   .WORD   <<GET_GRP_PRO1-U.271>-2>        ;
              7045  001EE ;TPA$TYPE
                               U.272:   .WORD   28741                           ;
         00000000*  001F0 ;TPA$ADDR
                               U.273:   .LONG   <<GROUP_PROT-U.273>-4>          ;
          00000004  001F4 ;TPA$MASK
                               U.274:   .LONG   4                               ;
             0000*  001F8 ;TPA$TARGET
                               U.275:   .WORD   <<GET_GRP_PRO1-U.275>-2>        ;
              7044  001FA ;TPA$TYPE
                               U.276:   .WORD   28740                           ;
         00000000*  001FC ;TPA$ADDR
                               U.277:   .LONG   <<GROUP_PROT-U.277>-4>          ;
          00000008  00200 ;TPA$MASK
                               U.278:   .LONG   8                               ;
             0000*  00204 ;TPA$TARGET
                               U.279:   .WORD   <<GET_GRP_PRO1-U.279>-2>        ;
              7043  00206 ;TPA$TYPE
                               U.280:   .WORD   28739                           ;
         00000000*  00208 ;TPA$ADDR
                               U.281:   .LONG   <<GROUP_PROT-U.281>-4>          ;
          00000010  0020C ;TPA$MASK
                               U.282:   .LONG   16                              ;
             0000*  00210 ;TPA$TARGET
                               U.283:   .WORD   <<GET_GRP_PRO1-U.283>-2>        ;
              15F6  00212 ;TPA$TYPE
                               U.284:   .WORD   5622                            ;
             0000*  00214 ;TPA$TARGET
                               U.285:   .WORD   <<U.210-U.285>-2>               ;
                    00216 ;GET_WOR_PRO
                               U.202:   .BLKB   0
              003A  00216 ;TPA$TYPE
                               U.286:   .WORD   58                              ;
              003D  00218 ;TPA$TYPE
                               U.287:   .WORD   61                              ;
              15F6  0021A ;TPA$TYPE
                               U.288:   .WORD   5622                            ;
             0000*  0021C ;TPA$TARGET
                               U.289:   .WORD   <<U.210-U.289>-2>               ;
                    0021E GET_WOR_PRO1:
                                        .BLKB   0
              7052  0021E ;TPA$TYPE
                               U.290:   .WORD   28754                           ;
         00000000*  00220 ;TPA$ADDR
                               U.291:   .LONG   <<WORLD_PROT-U.291>-4>          ;
          00000001  00224 ;TPA$MASK
                               U.292:   .LONG   1                               ;
             0000*  00228 ;TPA$TARGET
                               U.293:   .WORD   <<GET_WOR_PRO1-U.293>-2>        ;
              7057  0022A ;TPA$TYPE
                               U.294:   .WORD   28759                           ;
         00000000*  0022C ;TPA$ADDR
                               U.295:   .LONG   <<WORLD_PROT-U.295>-4>          ;
```

```
       00000002  00230 ;TPA$MASK
                       U.296:   .LONG    2                                   ;
          0000*  00234 ;TPA$TARGET
                       U.297:   .WORD    <<GET_WOR_PRO1-U.297>-2>            ;
           7045  00236 ;TPA$TYPE
                       U.298:   .WORD    28741                               ;
      00000000*  00238 ;TPA$ADDR
                       U.299:   .LONG    <<WORLD_PROT-U.299>-4>              ;
       00000004  0023C ;TPA$MASK
                       U.300:   .LONG    4                                   ;
          0000*  00240 ;TPA$TARGET
                       U.301:   .WORD    <<GET_WOR_PRO1-U.301>-2>            ;
           7044  00242 ;TPA$TYPE
                       U.302:   .WORD    28740                               ;
      00000000*  00244 ;TPA$ADDR
                       U.303:   .LONG    <<WORLD_PROT-U.303>-4>              ;
       00000008  00248 ;TPA$MASK
                       U.304:   .LONG    8                                   ;
          0000*  0024C ;TPA$TARGET
                       U.305:   .WORD    <<GET_WOR_PRO1-U.305>-2>            ;
           7043  0024E ;TPA$TYPE
                       U.306:   .WORD    28739                               ;
      00000000*  00250 ;TPA$ADDR
                       U.307:   .LONG    <<WORLD_PROT-U.307>-4>              ;
       00000010  00254 ;TPA$MASK
                       U.308:   .LONG    16                                  ;
          0000*  00258 ;TPA$TARGET
                       U.309:   .WORD    <<GET_WOR_PRO1-U.309>-2>            ;
           15F6  0025A ;TPA$TYPE
                       U.310:   .WORD    5622                                ;
          0000*  0025C ;TPA$TARGET
                       U.311:   .WORD    <<U.210-U.311>-2>                   ;
                 0025E ;CHK_END_PRO
                       U.210:   .BLKB    0
           102C  0025E ;TPA$TYPE
                       U.312:   .WORD    4140                                ;
          0000*  00260 ;TPA$TARGET
                       U.313:   .WORD    <<GET_PROT_CLASS-U.313>-2>          ;
           1429  00262 ;TPA$TYPE
                       U.314:   .WORD    5161                                ;
          0000*  00264 ;TPA$TARGET
                       U.315:   .WORD    <<U.73-U.315>-2>                    ;
                 00266 ;GET_STRING
                       U.91:    .BLKB    0
           102C  00266 ;TPA$TYPE
                       U.316:   .WORD    4140                                ;
           FFFE  00268 ;TPA$TARGET
                       U.317:   .WORD    -2                                  ;
           1029  0026A ;TPA$TYPE
                       U.318:   .WORD    4137                                ;
           FFFE  0026C ;TPA$TARGET
                       U.319:   .WORD    -2                                  ;
           11F7  0026E ;TPA$TYPE
                       U.320:   .WORD    4599                                ;
           FFFE  00270 ;TPA$TARGET
                       U.321:   .WORD    -2                                  ;
           0DF8  00272 ;TPA$TYPE
```

```
                  U.322:  .WORD   3576                                    ;
0000*  00274  ;TPA$SUBEXP
                  U.324:  .WORD   <<U.323-U.324>-2>                       ;
       00276  ;GET_STRING1
                  U.323:  .BLKB   0
19F8   00276  ;TPA$TYPE
                  U.325:  .WORD   6648                                    ;
0000*  00278  ;TPA$SUBEXP
                  U.327:  .WORD   <<U.326-U.327>-2>                       ;
0000*  0027A  ;TPA$TARGET
                  U.328:  .WORD   <<U.323-U.328>-2>                       ;
15F6   0027C  ;TPA$TYPE
                  U.329:  .WORD   5622                                    ;
FFFF   0027E  ;TPA$TARGET
                  U.330:  .WORD   -1                                      ;
       00280  ;CHK_DELIM
                  U.326:  .BLKB   0
102B   00280  ;TPA$TYPE
                  U.331:  .WORD   4139                                    ;
FFFE   00282  ;TPA$TARGET
                  U.332:  .WORD   -2                                      ;
102C   00284  ;TPA$TYPE
                  U.333:  .WORD   4140                                    ;
FFFE   00286  ;TPA$TARGET
                  U.334:  .WORD   -2                                      ;
1029   00288  ;TPA$TYPE
                  U.335:  .WORD   4137                                    ;
FFFE   0028A  ;TPA$TARGET
                  U.336:  .WORD   -2                                      ;
11F7   0028C  ;TPA$TYPE
                  U.337:  .WORD   4599                                    ;
FFFE   0028E  ;TPA$TARGET
                  U.338:  .WORD   -2                                      ;
15ED   00290  ;TPA$TYPE
                  U.339:  .WORD   5613                                    ;
FFFF   00292  ;TPA$TARGET
                  U.340:  .WORD   -1                                      ;
       00294  ;CHK_FOR_END
                  U.73:   .BLKB   0
15F7   00294  ;TPA$TYPE
                  U.341:  .WORD   5623                                    ;
FFFF   00296  ;TPA$TARGET
                  U.342:  .WORD   -1                                      ;

                          .PSECT  _LIB$KEY0$,NOWRT,  SHR,  PIC,1

       00000  ACE_KEY::
                          .BLKB   0
       00000  ;TPA$KEY0
                  U.1:    .BLKB   0
0000*  00000  ;TPA$KEY
                  U.4:    .WORD   <U.3-U.1>                               ;
0000*  00002  ;TPA$KEY
                  U.12:   .WORD   <U.11-U.1>                              ;
0000*  00004  ;TPA$KEY
                  U.20:   .WORD   <U.19-U.1>                              ;
0000*  00006  ;TPA$KEY
```

```
                                             U.27:     .WORD   <U.26-U.1>                              ;
                              0000*  00008   ;TPA$KEY
                                             U.34:     .WORD   <U.33-U.1>                              ;
                              0000*  0000A   ;TPA$KEY
                                             U.42:     .WORD   <U.41-U.1>                              ;
                              0000*  0000C   ;TPA$KEY
                                             U.50:     .WORD   <U.49-U.1>                              ;
                              0000*  0000E   ;TPA$KEY
                                             U.56:     .WORD   <U.55-U.1>                              ;
                              0000*  00010   ;TPA$KEY
                                             U.62:     .WORD   <U.61-U.1>                              ;
                              0000*  00012   ;TPA$KEY
                                             U.117:    .WORD   <U.116-U.1>                             ;
                              0000*  00014   ;TPA$KEY
                                             U.123:    .WORD   <U.122-U.1>                             ;
                              0000*  00016   ;TPA$KEY
                                             U.129:    .WORD   <U.128-U.1>                             ;
                              0000*  00018   ;TPA$KEY
                                             U.145:    .WORD   <U.144-U.1>                             ;
                              0000*  0001A   ;TPA$KEY
                                             U.151:    .WORD   <U.150-U.1>                             ;
                              0000*  0001C   ;TPA$KEY
                                             U.157:    .WORD   <U.156-U.1>                             ;
                              0000*  0001E   ;TPA$KEY
                                             U.163:    .WORD   <U.162-U.1>                             ;
                              0000*  00020   ;TPA$KEY
                                             U.169:    .WORD   <U.168-U.1>                             ;
                              0000*  00022   ;TPA$KEY
                                             U.181:    .WORD   <U.180-U.1>                             ;
                              0000*  00024   ;TPA$KEY
                                             U.187:    .WORD   <U.186-U.1>                             ;
                              0000*  00026   ;TPA$KEY
                                             U.193:    .WORD   <U.192-U.1>                             ;
                              0000*  00028   ;TPA$KEY
                                             U.199:    .WORD   <U.198-U.1>                             ;

                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

                     44  41  45  52  00000   P.AAC:    .ASCII  \READ\                                  ;
                          00000004   00004   P.AAB:    .LONG   4                                       ;
                          00000000'  00008             .ADDRESS P.AAC                                  ;
                 45  54  49  52  57  0000C   P.AAE:    .ASCII  \WRITE\                                 ;
                                     00011             .BLKB   3
                          00000005   00014   P.AAD:    .LONG   5                                       ;
                          00000000'  00018             .ADDRESS P.AAE                                  ;
         45  54  55  43  45  58  45  0001C   P.AAG:    .ASCII  \EXECUTE\                               ;
                                     00023             .BLKB   1
                          00000007   00024   P.AAF:    .LONG   7                                       ;
                          00000000'  00028             .ADDRESS P.AAG                                  ;
             45  54  45  4C  45  44  0002C   P.AAI:    .ASCII  \DELETE\                                ;
                                     00032             .BLKB   2
                          00000006   00034   P.AAH:    .LONG   6                                       ;
                          00000000'  00038             .ADDRESS P.AAI                                  ;
         4C  4F  52  54  4E  4F  43  0003C   P.AAK:    .ASCII  \CONTROL\                               ;
                                     00043             .BLKB   1
                          00000007   00044   P.AAJ:    .LONG   7                                       ;
                          00000000'  00048             .ADDRESS P.AAK                                  ;
         35  5F  54  49  42  0004C P.AAM:   .ASCII   \BIT_5\                          ;
                             00051          .BLKB    3                                ;
                   00000005  00054 P.AAL:   .LONG    5                                ;
                   00000000' 00058          .ADDRESS P.AAM                            ;
         36  5F  54  49  42  0005C P.AAO:   .ASCII   \BIT_6\                          ;
                             00061          .BLKB    3                                ;
                   00000005  00064 P.AAN:   .LONG    5                                ;
                   00000000' 00068          .ADDRESS P.AAO                            ;
         37  5F  54  49  42  0006C P.AAQ:   .ASCII   \BIT_7\                          ;
                             00071          .BLKB    3                                ;
                   00000005  00074 P.AAP:   .LONG    5                                ;
                   00000000' 00078          .ADDRESS P.AAQ                            ;
         38  5F  54  49  42  0007C P.AAS:   .ASCII   \BIT_8\                          ;
                             00081          .BLKB    3                                ;
                   00000005  00084 P.AAR:   .LONG    5                                ;
                   00000000' 00088          .ADDRESS P.AAS                            ;
         39  5F  54  49  42  0008C P.AAU:   .ASCII   \BIT_9\                          ;
                             00091          .BLKB    3                                ;
                   00000005  00094 P.AAT:   .LONG    5                                ;
                   00000000' 00098          .ADDRESS P.AAU                            ;
     30  31  5F  54  49  42  0009C P.AAW:   .ASCII   \BIT_10\                         ;
                             000A2          .BLKB    2                                ;
                   00000006  000A4 P.AAV:   .LONG    6                                ;
                   00000000' 000A8          .ADDRESS P.AAW                            ;
     31  31  5F  54  49  42  000AC P.AAY:   .ASCII   \BIT_11\                         ;
                             000B2          .BLKB    2                                ;
                   00000006  000B4 P.AAX:   .LONG    6                                ;
                   00000000' 000B8          .ADDRESS P.AAY                            ;
     32  31  5F  54  49  42  000BC P.ABA:   .ASCII   \BIT_12\                         ;
                             000C2          .BLKB    2                                ;
                   00000006  000C4 P.AAZ:   .LONG    6                                ;
                   00000000' 000C8          .ADDRESS P.ABA                            ;
     33  31  5F  54  49  42  000CC P.ABC:   .ASCII   \BIT_13\                         ;
                             000D2          .BLKB    2                                ;
                   00000006  000D4 P.ABB:   .LONG    6                                ;
                   00000000' 000D8          .ADDRESS P.ABC                            ;
     34  31  5F  54  49  42  000DC P.ABE:   .ASCII   \BIT_14\                         ;
                             000E2          .BLKB    2                                ;
                   00000006  000E4 P.ABD:   .LONG    6                                ;
                   00000000' 000E8          .ADDRESS P.ABE                            ;
     35  31  5F  54  49  42  000EC P.ABG:   .ASCII   \BIT_15\                         ;
                             000F2          .BLKB    2                                ;
                   00000006  000F4 P.ABF:   .LONG    6                                ;
                   00000000' 000F8          .ADDRESS P.ABG                            ;
     36  31  5F  54  49  42  000FC P.ABI:   .ASCII   \BIT_16\                         ;
                             00102          .BLKB    2                                ;
                   00000006  00104 P.ABH:   .LONG    6                                ;
                   00000000' 00108          .ADDRESS P.ABI                            ;
     37  31  5F  54  49  42  0010C P.ABK:   .ASCII   \BIT_17\                         ;
                             00112          .BLKB    2                                ;
                   00000006  00114 P.ABJ:   .LONG    6                                ;
                   00000000' 00118          .ADDRESS P.ABK                            ;
     38  31  5F  54  49  42  0011C P.ABM:   .ASCII   \BIT_18\                         ;
                             00122          .BLKB    2                                ;
                   00000006  00124 P.ABL:   .LONG    6                                ;
                   00000000' 00128          .ADDRESS P.ABM                            ;
     39  31  5F  54  49  42  0012C P.ABO:   .ASCII   \BIT_19\                         ;
```

```
                                                00132         .BLKB    2
                                     00000006   00134 P.ABN:  .LONG    6                                    ;
                                     00000000'  00138         .ADDRESS P.ABO                                ;
                     30  32  5F  54  49  42   0013C P.ABO:  .ASCII   \BIT_20\                             ;
                                                00142         .BLKB    2
                                     00000006   00144 P.ABP:  .LONG    6                                    ;
                                     00000000'  00148         .ADDRESS P.ABQ                                ;
                     31  32  5F  54  49  42   0014C P.ABS:  .ASCII   \BIT_21\                             ;
                                                00152         .BLKB    2
                                     00000006   00154 P.ABR:  .LONG    6                                    ;
                                     00000000'  00158         .ADDRESS P.ABS                                ;
                     32  32  5F  54  49  42   0015C P.ABU:  .ASCII   \BIT_22\                             ;
                                                00162         .BLKB    2
                                     00000006   00164 P.ABT:  .LONG    6                                    ;
                                     00000000'  00168         .ADDRESS P.ABU                                ;
                     33  32  5F  54  49  42   0016C P.ABW:  .ASCII   \BIT_23\                             ;
                                                00172         .BLKB    2
                                     00000006   00174 P.ABV:  .LONG    6                                    ;
                                     00000000'  00178         .ADDRESS P.ABW                                ;
                     34  32  5F  54  49  42   0017C P.ABY:  .ASCII   \BIT_24\                             ;
                                                00182         .BLKB    2
                                     00000006   00184 P.ABX:  .LONG    6                                    ;
                                     00000000'  00188         .ADDRESS P.ABY                                ;
                     35  32  5F  54  49  42   0018C P.ACA:  .ASCII   \BIT_25\                             ;
                                                00192         .BLKB    2
                                     00000006   00194 P.ABZ:  .LONG    6                                    ;
                                     00000000'  00198         .ADDRESS P.ACA                                ;
                     36  32  5F  54  49  42   0019C P.ACC:  .ASCII   \BIT_26\                             ;
                                                001A2         .BLKB    2
                                     00000006   001A4 P.ACB:  .LONG    6                                    ;
                                     00000000'  001A8         .ADDRESS P.ACC                                ;
                     37  32  5F  54  49  42   001AC P.ACE:  .ASCII   \BIT_27\                             ;
                                                001B2         .BLKB    2
                                     00000006   001B4 P.ACD:  .LONG    6                                    ;
                                     00000000'  001B8         .ADDRESS P.ACE                                ;
                     38  32  5F  54  49  42   001BC P.ACG:  .ASCII   \BIT_28\                             ;
                                                001C2         .BLKB    2
                                     00000006   001C4 P.ACF:  .LONG    6                                    ;
                                     00000000'  001C8         .ADDRESS P.ACG                                ;
                     39  32  5F  54  49  42   001CC P.ACI:  .ASCII   \BIT_29\                             ;
                                                001D2         .BLKB    2
                                     00000006   001D4 P.ACH:  .LONG    6                                    ;
                                     00000000'  001D8         .ADDRESS P.ACI                                ;
                     30  33  5F  54  49  42   001DC P.ACK:  .ASCII   \BIT_30\                             ;
                                                001E2         .BLKB    2
                                     00000006   001E4 P.ACJ:  .LONG    6                                    ;
                                     00000000'  001E8         .ADDRESS P.ACK                                ;
                     31  33  5F  54  49  42   001EC P.ACM:  .ASCII   \BIT_31\                             ;
                                                001F2         .BLKB    2
                                     00000006   001F4 P.ACL:  .LONG    6                                    ;
                                     00000000'  001F8         .ADDRESS P.ACM                                ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000'  001FC P.AAA:  .ADDRESS P.AAB, P.AAD, P.AAF, P.AAH, P.AAJ, -      ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000'  00214                  P.AAL, P.AAN, P.AAP, P.AAR, P.AAT, P.AAV, -  ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000'  0022C                  P.AAX, P.AAZ, P.ABB, P.ABD, P.ABF, P.ABH, -  ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000'  00244                  P.ABJ, P.ABL, P.ABN, P.ABP, P.ABR, P.ABT, -  ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000'  0025C                  P.ABV, P.ABX, P.ABZ, P.ACB, P.ACD, P.ACF, -  ;
                                        00000000' 00000000'  00274                  P.ACH, P.ACJ, P.ACL                          ;
```

```
                  4B  43  4F  4C  24  4C  43  41  0027C P.ACP:  .ASCII  \ACL$LOCK\                                    ;
                                        00000008  00284 P.ACO:  .LONG   8                                             ;
                                        00000000' 00288         .ADDRESS P.ACP                                        ;
                  5F  4C  49  46  24  4C  43  41  0028C P.ACR:  .ASCII  \ACL$FIL_\                                    ;
                                        00000008  00294 P.ACQ:  .LONG   8                                             ;
                                        00000000' 00298         .ADDRESS P.ACR                                        ;
                  5F  56  45  44  24  4C  43  41  0029C P.ACT:  .ASCII  \ACL$DEV_\                                    ;
                                        00000008  002A4 P.ACS:  .LONG   8                                             ;
                                        00000000' 002A8         .ADDRESS P.ACT                                        ;
                  5F  43  42  4A  24  4C  43  41  002AC P.ACV:  .ASCII  \ACL$JBC_\                                    ;
                                        00000008  002B4 P.ACU:  .LONG   8                                             ;
                                        00000000' 002B8         .ADDRESS P.ACV                                        ;
                  5F  46  45  43  24  4C  43  41  002BC P.ACX:  .ASCII  \ACL$CEF_\                                    ;
                                        00000008  002C4 P.ACW:  .LONG   8                                             ;
                                        00000000' 002C8         .ADDRESS P.ACX                                        ;
                  5F  54  4E  4C  24  4C  43  41  002CC P.ACZ:  .ASCII  \ACL$LNT_\                                    ;
                                        00000008  002D4 P.ACY:  .LONG   8                                             ;
                                        00000000' 002D8         .ADDRESS P.ACZ                                        ;
                  5F  43  52  50  24  4C  43  41  002DC P.ADB:  .ASCII  \ACL$PRC_\                                    ;
                                        00000008  002E4 P.ADA:  .LONG   8                                             ;
                                        00000000' 002E8         .ADDRESS P.ADB                                        ;
                  5F  4C  42  47  24  4C  43  41  002EC P.ADD:  .ASCII  \ACL$GBL_\                                    ;
                                        00000008  002F4 P.ADC:  .LONG   8                                             ;
                                        00000000' 002F8         .ADDRESS P.ADD                                        ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 002FC P.ACN:  .ADDRESS P.ACO, P.ACQ, P.ACS, P.ACU, P.ACW, -  ;
                                        00000000' 00000000' 00314                  P.ACY, P.ADA, P.ADC                  ;
                  59  54  49  52  55  43  45  53  0031C P.ADE:  .ASCII  \SECURITY\                                    ;

                                                                .PSECT  $OWN$,NOEXE,2

                                              00  00000 JOURNAL_ACES:
                                                                .BYTE   0                                             ;
                                                  00001         .BLKB   3
                                                  00004 ACE_BUFFER:
                                                                .BLKB   512
                                                  00204 ACE_INDEX:
                                                                .BLKB   4
                                                  00208 ACE_TYPE:
                                                                .BLKB   4
                                                  0020C ACE_RIGHTS:
                                                                .BLKB   4
                                                  00210 UIC_FLAGS:
                                                                .BLKB   4
                                                  00214 UIC_COUNT:
                                                                .BLKB   4
                                                  00218 IDENTIFIER:
                                                                .BLKB   4
                                                  0021C ID_NAME:.BLKB   8
                                                  00224 ID_COUNT:
                                                                .BLKB   4
                                                  00228 JOURNAL_NAME:
                                                                .BLKB   8
                                                  00230 ACCESS_FLAGS:
                                                                .BLKB   4
                                                  00234 SYSTEM_PROT:
                                                                .BLKB   4
                                                  00238 OWNER_PROT:
                                                         .BLKB    4
                                    0023C GROUP_PROT:
                                                         .BLKB    4
                                    00240 WORLD_PROT:
                                                         .BLKB    4
                                    00244 BIT_NAME_TABLE:
                                                         .BLKB    4
                                    00248 CHANGE_ACMODE:
                                                         .BLKB    4
                                    0024C CALL_ACMODE:
                                                         .BLKB    4
                                    00250 PARENT_ID:
                                                         .BLKB    4
                                    00254 ACL_QUEUE_HEAD:
                                                         .BLKB    4
                                    00258 ACL_POINTER:
                                                         .BLKB    4
                                    0025C ACL_SPLIT:
                                                         .BLKB    4
                                    00260 ACE_POINTER:
                                                         .BLKB    4
                                    00264 ACE_NUMBER:
                                                         .BLKB    4
                                    00268 ACL_AREA:
                                                         .BLKB    512
                                    00468 ACL_CONTEXT:
                                                         .BLKB    4
                                    0046C LOCK_RESNAM:
                                                         .BLKB    8
                                    00474 RESNAM_TEXT:
                                                         .BLKB    31

                                          DEFAULT_BITS=           P.AAA
                                          LOCK_PREFIX=            P.ACN
                                                         .EXTRN   ACL_ADDENTRY, ACL_DELENTRY
                                                         .EXTRN   ACL_MODENTRY, ACL_FINDENTRY
                                                         .EXTRN   ACL_FINDTYPE, ACL_DELETEACL
                                                         .EXTRN   ACL_READACL, ACL_ACLLENGTH
                                                         .EXTRN   ACL_READACE, ACL_LOCATEACE
                                                         .EXTRN   ACL_INIT_QUEUE, ALLOC_PAGED
                                                         .EXTRN   DALLOC_PAGED, LIB$TPARSE
                                                         .EXTRN   LIB$FID_TO_NAME
                                                         .EXTRN   LIB$GET_VM, LIB$FREE_VM
                                                         .EXTRN   EXE$PROBER, EXE$PROBEW
                                                         .EXTRN   IOC$VERIFYCHAN, SCH$LOCKR
                                                         .EXTRN   SCH$LOCKW, SCH$UNLOCK
                                                         .EXTRN   CTL$GL_PCB, LIB$_SYNTAXERR

                                                         .PSECT   $CODE$,NOWRT,2

                              007C 00000                 .ENTRY   SYS$PARSE_ACL, Save R2,R3,R4,R5,R6       ; 0616
             56 00000000'   EF   9E 00002                MOVAB    JOURNAL_NAME, R6                         ;
             5E             24   C2 00009                SUBL2    #36, SP                                  ;
                     1C     A6   D4 0000C                CLRL     BIT_NAME_TABLE                           ; 0676
             50      10     AC   D0 0000F                MOVL     BIT_TABLE, R0                            ; 0677
                            0C   13 00013                BEQL     1$                                       ;
   60     0100   8F         00   0C 00015                PROBER   #0, #256, (R0)                           ; 0678
```

```
                                              5B   13 0001B          BEQL    2$                                          :
                              1C   A6         50   D0 0001D          MOVL    R0, BIT_NAME_TABLE                          : 0679
  0200   8F         00              6E         00   2C 00021  1$:     MOVC5   #0, (SP), #0, #512, ACE_BUFFER              : 0684
                                        FDDC   C6      00028                                                             :
                                          E0   A6   7C 0002B          CLRQ    ACE_TYPE                                    : 0685
                                          DC   A6   D4 0002E          CLRL    ACE_INDEX                                   :
                                          E8   A6   7C 00031          CLRQ    UIC_FLAGS                                   : 0686
                                          FC   A6   D4 00034          CLRL    ID_COUNT                                    : 0687
                                          F4   A6   B4 00037          CLRW    ID_NAME                                     : 0689
                                               66   B4 0003A          CLRW    JOURNAL_NAME                                :
                                          14   A6   7C 0003C          CLRQ    GROUP_PROT                                  : 0690
                                          10   A6   D4 0003F          CLRL    OWNER_PROT                                  :
                                          08   A6   7C 00042          CLRQ    ACCESS_FLAGS                                : 0688
         24         00              6E         00   2C 00045          MOVC5   #0, (SP), #0, #36, TPARSE_BLOCK             : 0692
                                               6E      0004A                                                             :
                                    6E         08   D0 0004B          MOVL    #8, TPARSE_BLOCK                            : 0693
                              04    AE         02   88 0004E          BISB2   #2, TPARSE_BLOCK+4                          : 0694
                                    54    04   AC   D0 00052          MOVL    ACL_STRING, R4                              : 0695
                         64         08         00   0C 00056          PROBER  #0, #8, (R4)                                :
                                               1C   13 0005A          BEQL    2$                                          :
                                    50         64   3C 0005C          MOVZWL  (R4), R0                                    : 0698
                              08    AE         50   D0 0005F          MOVL    R0, TPARSE_BLOCK+8                          :
                                    55         50   D0 00063          MOVL    R0, ACL_STRING_LEN                          :
                                    50    04   A4   D0 00066          MOVL    4(R4), R0                                   : 0699
                                    51         55   D0 0006A          MOVL    ACL_STRING_LEN, R1                          :
                                               53   D4 0006D          CLRL    R3                                          :
                                   00000000G   00   16 0006F          JSB     EXE$PROBER                                  :
                                    03         50   E8 00075          BLBS    R0, 3$                                      :
                                        017E   31      00078  2$:     BRW     17$                                         :
                              0C    AE    04   A4   D0 0007B  3$:     MOVL    4(R4), TPARSE_BLOCK+12                      : 0700
                                   00000000'   EF   9F 00080          PUSHAB  ACE_KEY                                     : 0705
                                   00000000'   EF   9F 00086          PUSHAB  ACE_STATE                                   :
                                          08   AE   9F 0008C          PUSHAB  TPARSE_BLOCK                                :
                   00000000G  00               03   FB 0008F          CALLS   #3, LIB$TPARSE                              :
                                    51    0C   AC   D0 00096          MOVL    ERROR_POSITION, R1                          : 0709
                                               0B   13 0009A          BEQL    4$                                          :
                         61         02         00   0D 0009C          PROBEW  #0, #2, (R1)                                : 0710
                                               D6   13 000A0          BEQL    2$                                          :
                         61         55    08   AE   A3 000A2          SUBW3   TPARSE_BLOCK+8, ACL_STRING_LEN, (R1)        : 0712
                   00000000G  8F               50   D1 000A7  4$:     CMPL    STATUS, #LIB$_SYNTAXERR                     : 0718
                                               44   13 000AE          BEQL    8$                                          :
                                          E0   A6   D5 000B0          TSTL    ACE_TYPE                                    : 0719
                                               3F   13 000B3          BEQL    8$                                          :
                                    05         50   E9 000B5          BLBC    STATUS, 5$                                  : 0720
                                          08   AE   D5 000B8          TSTL    TPARSE_BLOCK+8                              :
                                               37   14 000BB          BGTR    8$                                          :
                                    01         50   E8 000BD  5$:     BLBS    STATUS, 6$                                  : 0723
                                                    04 000C0          RET                                                 :
                        FDDD  C6          E0   A6   90 000C1  6$:     MOVB    ACE_TYPE, ACE_BUFFER+1                      : 0727
                              50        FDDE   C6   3C 000C7          MOVZWL  ACE_BUFFER+2, R0                            : 0729
                              50          E8   A6   C8 000CC          BISL2   UIC_FLAGS, R0                               :
          FDDE  C6            50          08   A6   A9 000D0          BISW3   ACCESS_FLAGS, R0, ACE_BUFFER+2              : 0730
                        FDE0  C6          E4   A6   D0 000D7          MOVL    ACE_RIGHTS, ACE_BUFFER+4                    : 0731
                              08          01                                                                             :
                              E0          A6   CF      000DD          CASEL   ACE_TYPE, #1, #8                            : 0736
  0039           0039               0039    0015      000E2  7$:     .WORD   9$-7$,-                                     :
  009F           009F               0061    0061      000EA                  10$-7$,-                                    :
                                            008C      000F2                  10$-7$,-                                    :
```

```
                                                                         10$-7$,-                                    ;
                                                                         11$-7$,-                                    ;
                                                                         11$-7$,-                                    ;
                                                                         14$-7$,-                                    ;
                                                                         14$-7$,-                                    ;
                                                                         13$-7$                                      ;
                                            008A  31 000F4 8$:      BRW   14$                                        ; 0799
                                      08    A6    D5 000F7 9$:      TSTL  ACCESS_FLAGS                               ; 0740
                                            F8    12 000FA          BNEQ  8$                                         ;
                                            66    B5 000FC          TSTW  JOURNAL_NAME                               ; 0741
                                            F4    12 000FE          BNEQ  8$                                         ;
                        01          EC      A6    D1 00100          CMPL  UIC_COUNT, #1                              ; 0742
                                            7B    14 00104          BGTR  14$                                        ;
                                    DC      A6    D5 00106          TSTL  ACE_INDEX                                  ; 0743
                                            76    13 00109          BEQL  14$                                        ;
                        50          DC      A6    D0 0010B          MOVL  ACE_INDEX, R0                              ; 0745
            51          50                  02    78 0010F          ASHL  #2, R0, R1                                 ;
       FDDC C6          51                  08    81 00113          ADDB3 #8, R1, ACE_BUFFER                         ;
                                            51    11 00119          BRB   12$                                        ; 0736
                        61        FDD8      C6    E9 0011B 10$:     BLBC  JOURNAL_ACES, 14$                          ; 0751
                                    EC      A6    D5 00120          TSTL  UIC_COUNT                                  ; 0752
                                            5C    12 00123          BNEQ  14$                                        ;
                                    FC      A6    D5 00125          TSTL  ID_COUNT                                   ; 0753
                                            57    12 00128          BNEQ  14$                                        ;
                                    08      A6    D5 0012A          TSTL  ACCESS_FLAGS                               ; 0754
                                            52    12 0012D          BNEQ  14$                                        ;
                                    E4      A6    D5 0012F          TSTL  ACE_RIGHTS                                 ; 0755
                                            4D    12 00132          BNEQ  14$                                        ;
       FDE0 C6          04  B6              66    28 00134          MOVC3 JOURNAL_NAME, @JOURNAL_NAME+4, ACE_BUFFER+4 ; 0759
       FDDC C6              66              04    81 0013B          ADDB3 #4, JOURNAL_NAME, ACE_BUFFER               ; 0760
                                            29    11 00141          BRB   12$                                        ; 0736
                                    EC      A6    D5 00143 11$:     TSTL  UIC_COUNT                                  ; 0766
                                            39    12 00146          BNEQ  14$                                        ;
                                    FC      A6    D5 00148          TSTL  ID_COUNT                                   ; 0767
                                            34    12 0014B          BNEQ  14$                                        ;
                        08                  66    B1 0014D          CMPW  JOURNAL_NAME, #8                           ; 0768
                                            2F    12 00150          BNEQ  14$                                        ;
   66               00 00000000'  EF        08    2D 00152          CMPC5 #8, P.ADE, #0, JOURNAL_NAME, -             ; 0769
                                    04      B6       0015B                @JOURNAL_NAME+4                            ;
                                            22    12 0015D          BNEQ  14$                                        ;
       FDE4 C6          04  B6              66    28 0015F          MOVC3 JOURNAL_NAME, @JOURNAL_NAME+4, ACE_BUFFER+8 ; 0774
       FDDC C6              66              08    81 00166          ADDB3 #8, JOURNAL_NAME, ACE_BUFFER               ; 0775
                                            56    11 0016C 12$:     BRB   16$                                        ; 0736
                                    08      A6    D5 0016E 13$:     TSTL  ACCESS_FLAGS                               ; 0779
                                            0E    12 00171          BNEQ  14$                                        ;
                                            66    B5 00173          TSTW  JOURNAL_NAME                               ; 0780
                                            0A    12 00175          BNEQ  14$                                        ;
                                    EC      A6    D5 00177          TSTL  UIC_COUNT                                  ; 0781
                                            05    12 0017A          BNEQ  14$                                        ;
                                    FC      A6    D5 0017C          TSTL  ID_COUNT                                   ; 0782
                                            06    13 0017F          BEQL  15$                                        ;
                        50        21E4      8F    3C 00181 14$:     MOVZWL #8676, R0                                 ; 0783
                                                  04 00186          RET                                              ;
                    0C  A6          0C      A6    D2 00187 15$:     MCOML SYSTEM_PROT, SYSTEM_PROT                   ; 0784
   0C  A6       1B      05                  00    F0 0018C          INSV  #0, #5, #27, SYSTEM_PROT                   ; 0785
                    10  A6          10      A6    D2 00192          MCOML OWNER_PROT, OWNER_PROT                     ; 0787
   10  A6       1B      05                  00    F0 00197          INSV  #0, #5, #27, OWNER_PROT                    ; 0788
```

```
                                 FDE4   C6      0C   A6  7D 0019D           MOVQ    SYSTEM_PROT, ACE_BUFFER+8                    ; 0786
                                   14   A6      14   A6  D2 001A3           MCOML   GROUP_PROT, GROUP_PROT                       ; 0790
     14   A6             1B             05           00  F0 001A8           INSV    #0, #5, #27, GROUP_PROT                      ; 0791
                                   18   A6      18   A6  D2 001AE           MCOML   WORLD_PROT, WORLD_PROT                       ; 0793
     18   A6             1B             05           00  F0 001B3           INSV    #0, #5, #27, WORLD_PROT                      ; 0794
                                 FDEC   C6      14   A6  7D 001B9           MOVQ    GROUP_PROT, ACE_BUFFER+16                    ; 0792
                                 FDDC   C6           18  90 001BF           MOVB    #24, ACE_BUFFER                              ; 0796
                                        54      08   AC  D0 001C4 16$:      MOVL    ACL_ENTRY, R4                                ; 0804
                         64             08           00  0C 001C8           PROBER  #0, #8, (R4)                                 ;
                                        2B               13 001CC           BEQL    17$                                          ;
                                        51           64  3C 001CE           MOVZWL  (R4), ACL_ENTRY_LEN                          ; 0807
          51      FDDC   C6             08           00  ED 001D1           CMPZV   #0, #8, ACE_BUFFER, ACL_ENTRY_LEN            ; 0808
                                        1F               1A 001D8           BGTRU   17$                                          ;
                         50                     04   A4  D0 001DA           MOVL    4(R4), R0                                    ; 0809
                                        53               D4 001DE           CLRL    R3                                           ;
                                  00000000G          00  16 001E0           JSB     EXE$PROBEW                                   ;
                                        10           50  E9 001E6           BLBC    R0, 17$                                      ;
                                        50    FDDC   C6  9A 001E9           MOVZBL  ACE_BUFFER, R0                               ; 0810
          64             00      FDDC   C6           50  2C 001EE           MOVC5   R0, ACE_BUFFER, #0, (R4), @4(R4)             ; 0811
                                                04   B4     001F5                                                                ;
                                                     04  11 001F7           BRB     18$                                          ; 0810
                         50                          0C  D0 001F9 17$:      MOVL    #12, R0                                      ; 0814
                                                         04 001FC           RET                                                  ;
                         50                          01  D0 001FD 18$:      MOVL    #1, R0                                       ; 0816
                                                         04 00200           RET                                                  ; 0818

; Routine Size:  513 bytes,    Routine Base:  $CODE$ + 0000
;  823          0819  1 %SBTTL  '$FORMAT_ACL system service'
;  824          0820  1 GLOBAL ROUTINE SYS$FORMAT_ACL (ACL_ENTRY, ACL_LENGTH, ACL_STRING,
;  825          0821  1                                LINE_WIDTH, TERM_DESC, LINE_INDENT,
;  826          0822  1                                BIT_TABLE) =
;  827          0823  1
;  828          0824  1 !++
;  829          0825  1 !
;  830          0826  1 ! FUNCTIONAL DESCRIPTION:
;  831          0827  1 !
;  832          0828  1 !       This routine converts the Access Control Entry from a binary form
;  833          0829  1 !       to a text form.
;  834          0830  1 !
;  835          0831  1 ! CALLING SEQUENCE:
;  836          0832  1 !       SYS$FORMAT_ACL (ARG1, ARG2, ARG3, ARG4, ARG5, ARG6, ARG7)
;  837          0833  1 !
;  838          0834  1 ! INPUT PARAMETERS:
;  839          0835  1 !       ARG1: address of the input buffer descriptor
;  840          0836  1 !       ARG4: address of the maximum line width for formatting
;  841          0837  1 !       ARG5: address of the output line segment terminator descriptor
;  842          0838  1 !       ARG6: address of the number of columns to indent each line segment
;  843          0839  1 !       ARG7: address of an access bit name table
;  844          0840  1 !
;  845          0841  1 ! IMPLICIT INPUTS:
;  846          0842  1 !       none
;  847          0843  1 !
;  848          0844  1 ! OUTPUT PARAMETERS:
;  849          0845  1 !       ARG2: address of a word to get the length of the formatted ACE
;  850          0846  1 !       ARG3: address of the output text buffer descriptor
;  851          0847  1 !
;  852          0848  1 ! IMPLICIT OUTPUTS:
;  853          0849  1 !       none
;  854          0850  1 !
;  855          0851  1 ! ROUTINE VALUE:
;  856          0852  1 !       SS$_NORMAL:     The conversion was successful.
;  857          0853  1 !       SS$_NOSUCHID:   The identifier specified in the ACE is not in the
;  858          0854  1 !                       rights database.
;  859          0855  1 !       SS$_BUFFEROVF:  The conversion was successful.  The formatted ACE
;  860          0856  1 !                       has overflowed the output buffer and has been
;  861          0857  1 !                       truncated.
;  862          0858  1 !
;  863          0859  1 ! SIDE EFFECTS:
;  864          0860  1 !       none
;  865          0861  1 !
;  866          0862  1 !--
;  867          0863  1
;  868          0864  2 BEGIN
;  869          0865  2
;  870        M 0866  2 MACRO   CHECK_WIDTH (TEST_WIDTH) =
;  871        M 0867  2                 BEGIN
;  872        M 0868  2                 IF .WIDTH GTRU 0
;  873        M 0869  2                 AND .LINE_SIZE + TEST_WIDTH GTRU .WIDTH
;  874        M 0870  2                 THEN
;  875        M 0871  2                     BEGIN
;  876        M 0872  2                     IF .TERM_LENGTH GTR 0
;  877        M 0873  2                     THEN
;  878        M 0874  2                         BEGIN
;  879        M 0875  2                         CH$MOVE (.TERM_LENGTH, .TERM_POINTER, BUFFER[.SIZE]);
```

```
                                SIZE = .SIZE + .TERM_DESC[DSC$W_LENGTH];
                                END;
                            CH$FILL (%C' ', .INDENT, BUFFER[.SIZE]);
                            SIZE = .SIZE + .INDENT;
                            LINE_SIZE = .INDENT;
                            END;
                        LINE_SIZE = .LINE_SIZE + TEST_WIDTH;
                        END
                        %,

                STORE_TEXT (STRING) =
                        BEGIN
                        CHECK_WIDTH (%CHARCOUNT (STRING));
                        CH$MOVE (%CHARCOUNT (STRING), UPLIT (STRING), BUFFER[.SIZE]);
                        SIZE = .SIZE + %CHARCOUNT (STRING);
                        END
                        %,

                NEW_LINE =
                        BEGIN
                        IF .TERM_LENGTH GTR 0
                        THEN
                            BEGIN
                            CH$MOVE (.TERM_LENGTH, .TERM_POINTER, BUFFER[.SIZE]);
                            SIZE = .SIZE + .TERM_LENGTH;
                            END;
                        CH$FILL (%C' ', .INDENT, BUFFER[.SIZE]);
                        SIZE = .SIZE + .INDENT;
                        LINE_SIZE = .INDENT;
                        END
                        %;

        MAP
                ACL_ENTRY       : REF $BBLOCK,          ! Address of the input descriptor
                ACL_STRING      : REF $BBLOCK,          ! Address of the output descriptor
                TERM_DESC       : REF $BBLOCK;          ! Segment terminator descriptor

        LITERAL
                MAX_FAO_LENGTH  = MAXU (2 * KGB$S_NAME + 3,     ! Max size of
                                        ATR$S_FILE_SPEC),       !   FAO buffer
                MAX_FMT_ACE     = 3072,                 ! Largest possible formatted ACE
                VOLNAM_SIZE     = %CHARCOUNT ('DISK$') + ACE$S_VOLNAM + 1;
                                                        ! Size of full volume name

        LOCAL
                ACL_ENTRY_LEN,                          ! Length of input ACE buffer
                LOCAL_ACE       : $BBLOCK [ATR$S_ADDACLENT],    ! Local copy of ACE
                FAO_DESCR       : $BBLOCK [DSC$C_S_BLN],        ! FAO output descriptor
                KEY_IDENTIFIER  : $BBLOCK [4],          ! Key identifier
                PROT_VALUE,                             ! Protection value from ACE
                PROT_FIELD_DSC  : REF $BBLOCK,          ! Addr of protection field name
                PROT_BUF        : VECTOR [32,BYTE],     ! Storage for ASCII protection string
                PROT_IDX,                               ! Index into protection string
                BUFFER          : VECTOR [MAX_FMT_ACE, BYTE],
                                                        ! Temp storage for formatted ACE
                LINE_SIZE,                              ! Size of the current segment
                SIZE,                                   ! Size of formatted ACE
```

```
2               INDENT,                                         ! Number of columns to indent
2               WIDTH,                                          ! Width of the line
2               TERM_LENGTH,                                    ! Size of terminator string
2               TERM_POINTER,                                   ! Address of terminator string
2               FAO_DESC        : $BBLOCK [DSC$C_S_BLN],        ! FAO output descriptor
2               FAO_BUF         : VECTOR [MAX_FAO_LENGTH,BYTE], ! FAO output buffer
2               BIT_NAME_DESC   : REF $BBLOCK,                  ! Descr for access bit name
2               FLAGS           : BITVECTOR [16],               ! Flags from ACE
2               ACCESS_MASK     : BYTE,                         ! Access mask in ACE
2               AUDIT_MASK      : BYTE,                         ! Audit access mask in ACE
2               VOLNAM_DESC     : $BBLOCK [DSC$C_S_BLN],        ! Volume name descriptor
2               VOLNAM_TEXT     : VECTOR [VOLNAM_SIZE, BYTE],   ! Volume name storage
2               FILENAME_DESC   : $BBLOCK [DSC$C_S_BLN],        ! File name descriptor
2               FILENAME_TEXT   : VECTOR [ATR$S_FILE_SPEC],     ! File name storage
2               ACL_STRING_LEN,                                 ! Length of ACL string buffer
2               LOCAL_STATUS;                                   ! Local routine return status
2
2       ! Protection code names.
2
2       BIND
2               PROT_CODE       = UPLIT BYTE ('R', 'W', 'E', 'D', 'C',
2                                       REP 27 OF (0)) : VECTOR [, BYTE];
2
2       ! Probe the output string buffer.
2
2       IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_STRING)
2       THEN
3           BEGIN
3           ACL_STRING_LEN = .ACL_STRING[DSC$W_LENGTH];
3           IF EXE$PROBEW (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
3           THEN CH$FILL (%C' ', .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
3           ELSE RETURN SS$_ACCVIO;
3           END
2       ELSE RETURN SS$_ACCVIO;
2
2       ! Set up initial parameters.
2
2       INDENT = WIDTH = 0;
2       TERM_LENGTH = TERM_POINTER = 0;
2       ACCESS_MASK = AUDIT_MASK = 0;
2
2       ! Check the optional arguments.
2
2       IF .LINE_WIDTH NEQA 0
2       THEN IF PROBER (%REF (0), %REF (4), .LINE_WIDTH)
2            THEN WIDTH = ..LINE_WIDTH
2            ELSE RETURN SS$_ACCVIO;
2
2       IF .TERM_DESC NEQA 0
2       THEN
3           BEGIN
3           IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .TERM_DESC)
3           THEN
4               BEGIN
4               TERM_LENGTH = .TERM_DESC[DSC$W_LENGTH];
4               TERM_POINTER = .TERM_DESC[DSC$A_POINTER];
4               IF NOT EXE$PROBER (0, .TERM_LENGTH, .TERM_POINTER)
```

```
4               THEN RETURN SS$_ACCVIO;
4               END
3           ELSE RETURN SS$_ACCVIO;
2           END;
2
2   IF .LINE_INDENT NEQA 0
2   THEN IF PROBER (%REF (0), %REF (4), .LINE_INDENT)
2            THEN INDENT = ..LINE_INDENT
2            ELSE RETURN SS$_ACCVIO;
2
2   IF .INDENT GTRU 0
2   THEN
3       BEGIN
3       IF .WIDTH GTR 0
4       THEN (IF .INDENT GTRU .WIDTH THEN RETURN SS$_BADPARAM)
3       ELSE (IF .INDENT GTRU .ACL_STRING_LEN THEN RETURN SS$_BUFFEROVF);
3       IF .INDENT GTRU MAX_FMT_ACE THEN RETURN SS$_BUFFEROVF;
2       END;
2
2
2   ! Check to see if an access bit name table was supplied.  If so, use it
2   ! rather than the default table.
2
2   BIT_NAME_TABLE = 0;
2   IF .BIT_TABLE NEQA 0
2   THEN IF PROBER (%REF (0), %REF (256), .BIT_TABLE)
2            THEN BIT_NAME_TABLE = .BIT_TABLE
2            ELSE RETURN SS$_ACCVIO;
2
2   ! Start building the text ACE.
2
2   CH$FILL (%C' ', .INDENT, BUFFER);
2   SIZE = LINE_SIZE = .INDENT;
2   STORE_TEXT ('(');
2   IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_ENTRY)
2   THEN
3       BEGIN
3       ACL_ENTRY_LEN = .ACL_ENTRY[DSC$W_LENGTH];
3       IF .ACL_ENTRY_LEN GTRU ATR$S_ADDACLENT THEN RETURN SS$_IVACL;
3       IF EXE$PROBER (0, .ACL_ENTRY_LEN, .ACL_ENTRY[DSC$A_POINTER])
3       THEN CH$MOVE (.ACL_ENTRY_LEN, .ACL_ENTRY[DSC$A_POINTER], LOCAL_ACE)
3       ELSE RETURN SS$_ACCVIO;
3       END
2   ELSE RETURN SS$_ACCVIO;
2
2   ! Convert the ACE type code.
2
2   CASE .LOCAL_ACE[ACE$B_TYPE] FROM ACE$C_KEYID TO ACE$C_DIRDEF OF
2   SET
3       [ACE$C_KEYID]:
3           BEGIN
3           ACCESS_MASK = 1;
3           STORE_TEXT ('IDENTIFIER=');
3           INCR J FROM .LOCAL_ACE[ACE$V_RESERVED] + 1 TO (.LOCAL_ACE[ACE$B_SIZE] - ACE$C_LENGTH + 3) / 4
3           DO
4               BEGIN
4               KEY_IDENTIFIER = .VECTOR [LOCAL_ACE[ACE$L_KEY], .J - 1];
```

```
    4              FAO_DESC[DSC$W_LENGTH] = MAX_FAO_LENGTH;       ! Max size of an identifier
    4              FAO_DESC[DSC$A_POINTER] = FAO_BUF;
P   4              $FAOL (CTRSTR = $DESCRIPTOR ('!%I'),
P   4                      OUTLEN = FAO_DESC,
P   4                      OUTBUF = FAO_DESC,
    4                      PRMLST = KEY_IDENTIFIER);
    4              CHECK_WIDTH (.FAO_DESC[DSC$W_LENGTH]);
    4              CH$MOVE (.FAO_DESC[DSC$W_LENGTH],
    4                       .FAO_DESC[DSC$A_POINTER],
    4                       BUFFER[.SIZE]);
    4              SIZE = .SIZE + .FAO_DESC[DSC$W_LENGTH];
    4              STORE_TEXT ('+');
    3              END;
    3          BUFFER[.SIZE - 1] = %C',';
    2          END;
    2      [ACE$C_BIJNL,
    2       ACE$C_AIJNL,
    2       ACE$C_ATJNL]:
    3          BEGIN
    3          IF .LOCAL_ACE[ACE$B_TYPE] EQL ACE$C_BIJNL
    3          THEN STORE_TEXT ('BI_JOURNAL=');
    3          IF .LOCAL_ACE[ACE$B_TYPE] EQL ACE$C_AIJNL
    3          THEN STORE_TEXT ('AI_JOURNAL=');
    3          IF .LOCAL_ACE[ACE$B_TYPE] EQL ACE$C_ATJNL
    3          THEN STORE_TEXT ('AT_JOURNAL=');
    3          CHECK_WIDTH (.LOCAL_ACE[ACE$B_SIZE] - $BYTEOFFSET (ACE$L_ACCESS));
    3          CH$MOVE (.LOCAL_ACE[ACE$B_SIZE] - $BYTEOFFSET (ACE$L_ACCESS),
    3                   LOCAL_ACE[ACE$L_ACCESS],
    3                   BUFFER[.SIZE]);
    3          SIZE = .SIZE + .LOCAL_ACE[ACE$B_SIZE] - $BYTEOFFSET (ACE$L_ACCESS);
    3          STORE_TEXT (',');
    2          END;
    2      [ACE$C_AUDIT,
    2       ACE$C_ALARM]:
    3          BEGIN
    3          ACCESS_MASK = 1;
    3          AUDIT_MASK = 1;
    3          IF .LOCAL_ACE[ACE$B_TYPE] EQL ACE$C_AUDIT
    3          THEN STORE_TEXT ('AUDIT_JOURNAL=');
    3          IF .LOCAL_ACE[ACE$B_TYPE] EQL ACE$C_ALARM
    3          THEN STORE_TEXT ('ALARM_JOURNAL=');
    3          CHECK_WIDTH (.LOCAL_ACE[ACE$B_SIZE] - ACE$C_LENGTH);
    3          CH$MOVE (.LOCAL_ACE[ACE$B_SIZE] - ACE$C_LENGTH,
    3                   LOCAL_ACE[ACE$L_KEY],
    3                   BUFFER[.SIZE]);
    3          SIZE = .SIZE + .LOCAL_ACE[ACE$B_SIZE] - ACE$C_LENGTH;
    3          STORE_TEXT (',');
    2          END;
    2      [ACE$C_DIRDEF]:
    3          BEGIN
    3          STORE_TEXT ('DEFAULT_PROTECTION,');
    3          INCR K FROM 0 TO 3
    3          DO
    4              BEGIN
    4              CASE .K FROM 0 TO 3 OF
    4              SET
    5                  [0]:    BEGIN
```

```
                    PROT_VALUE = .LOCAL_ACE[ACE$L_SYS_PROT];
                    PROT_FIELD_DSC = $DESCRIPTOR ('SYSTEM:');
                    END;
            [1]:    BEGIN
                    PROT_VALUE = .LOCAL_ACE[ACE$L_OWN_PROT];
                    PROT_FIELD_DSC = $DESCRIPTOR ('OWNER:');
                    END;
            [2]:    BEGIN
                    PROT_VALUE = .LOCAL_ACE[ACE$L_GRP_PROT];
                    PROT_FIELD_DSC = $DESCRIPTOR ('GROUP:');
                    END;
            [3]:    BEGIN
                    PROT_VALUE = .LOCAL_ACE[ACE$L_WOR_PROT];
                    PROT_FIELD_DSC = $DESCRIPTOR ('WORLD:');
                    END;
        TES;
        PROT_IDX = 0;
        INCR J FROM 0 TO 31
        DO
            BEGIN
            IF .PROT_CODE[.J] NEQ 0 AND NOT .PROT_VALUE<.J, 1>
            THEN
                BEGIN
                PROT_BUF[.PROT_IDX] = .PROT_CODE[.J];
                PROT_IDX = .PROT_IDX + 1;
                END;
            END;
        CHECK_WIDTH (.PROT_FIELD_DSC[DSC$W_LENGTH] + .PROT_IDX);
        CH$COPY (.PROT_FIELD_DSC[DSC$W_LENGTH], .PROT_FIELD_DSC[DSC$A_POINTER],
                .PROT_IDX, PROT_BUF,
                0,
                512 - .SIZE,
                BUFFER[.SIZE]);
        SIZE = .SIZE + .PROT_FIELD_DSC[DSC$W_LENGTH] + .PROT_IDX;
        STORE_TEXT (',');
        END;
    END;
[ACE$C_JNLID]:
    BEGIN
    STORE_TEXT ('RMS_JOURNAL_ID,');
    CH$FILL (0, DSC$C_S_BLN, VOLNAM_DESC);
    CH$FILL (0, VOLNAM_SIZE, VOLNAM_TEXT);
    CH$FILL (0, DSC$C_S_BLN, FILENAME_DESC);
    CH$FILL (0, ATR$S_FILE_SPEC, FILENAME_TEXT);
    CH$COPY (%CHARCOUNT ('DISK$'), UPLIT ('DISK$'),
            ACE$S_VOLNAM, LOCAL_ACE[ACE$T_VOLNAM],
            0, VOLNAM_SIZE, VOLNAM_TEXT);
    VOLNAM_DESC[DSC$W_LENGTH] = CH$FIND_CH (VOLNAM_SIZE, VOLNAM_TEXT, 0) -
                                VOLNAM_TEXT;
    VOLNAM_DESC[DSC$A_POINTER] = VOLNAM_TEXT;
    FILENAME_DESC[DSC$W_LENGTH] = ATR$S_FILE_SPEC;
    FILENAME_DESC[DSC$A_POINTER] = FILENAME_TEXT;
    LOCAL_STATUS = LIB$FID_TO_NAME (VOLNAM_DESC, LOCAL_ACE[ACE$T_FID],
                                    FILENAME_DESC, FILENAME_DESC);
    STORE_TEXT ('JOURNALED_FILE=');
    IF .LOCAL_STATUS
    THEN
```

```
     4                  BEGIN
     4                  LOCAL           SEGMENT_START   : REF VECTOR [,BYTE],
     4                                  SEGMENT_SIZE;           ! Size of segment to get
     4                  SEGMENT_START = .FILENAME_DESC[DSC$A_POINTER];
     4                  SEGMENT_SIZE = MINU (.WIDTH - .LINE_SIZE, .FILENAME_DESC[DSC$W_LENGTH]);
     4                  DO
     5                      BEGIN
     5                      IF .SEGMENT_SIZE LSSU .FILENAME_DESC[DSC$W_LENGTH]
     5                      THEN
     5                          DECR J FROM .SEGMENT_SIZE TO 1
     5                          DO
     6                              BEGIN
     6                              IF .SEGMENT_START[.J - 1] EQL ':'
     6                              OR .SEGMENT_START[.J - 1] EQL ']'
     6                              OR .SEGMENT_START[.J - 1] EQL '.'
     6                              OR .SEGMENT_START[.J - 1] EQL ';'
     6                              THEN
     7                                  BEGIN
     7                                  SEGMENT_SIZE = .J;
     7                                  EXITLOOP;
     6                                  END;
     5                              END;
     5                      CH$MOVE (.SEGMENT_SIZE, .SEGMENT_START, BUFFER[.SIZE]);
     5                      LINE_SIZE = .LINE_SIZE + .SEGMENT_SIZE;
     5                      SIZE = .SIZE + .SEGMENT_SIZE;
     5                      FILENAME_DESC[DSC$W_LENGTH] = .FILENAME_DESC[DSC$W_LENGTH] - .SEGMENT_SIZE;
     5                      SEGMENT_START = .SEGMENT_START + .SEGMENT_SIZE;
     5                      IF .FILENAME_DESC[DSC$W_LENGTH] GTR 0 THEN NEW_LINE;
     5                      SEGMENT_SIZE = MINU (.WIDTH - .LINE_SIZE, .FILENAME_DESC[DSC$W_LENGTH]);
     5                      END
     4                  UNTIL .FILENAME_DESC[DSC$W_LENGTH] LEQ 0;
     4                  STORE_TEXT (',');
     4                  END
     3              ELSE
     4                  BEGIN
     4                  FAO_DESC[DSC$W_LENGTH] = MAX_FAO_LENGTH;
     4                  FAO_DESC[DSC$A_POINTER] = FAO_BUF;
P    4                  $FAO ($DESCRIPTOR ('(!UW,!UW,!UW),'),
P    4                          FAO_DESC,
P    4                          FAO_DESC,
P    4                          .(LOCAL_ACE[ACE$T_FID] + $BYTEOFFSET (FID$W_NUM)),
P    4                          .(LOCAL_ACE[ACE$T_FID] + $BYTEOFFSET (FID$W_SEQ)),
     4                          .(LOCAL_ACE[ACE$T_FID] + $BYTEOFFSET (FID$W_RVN)));
     4                  CHECK_WIDTH (.FAO_DESC[DSC$W_LENGTH]);
     4                  CH$MOVE (.FAO_DESC[DSC$W_LENGTH], .FAO_DESC[DSC$A_POINTER], BUFFER[.SIZE]);
     4                  SIZE = .SIZE + .FAO_DESC[DSC$W_LENGTH];
     3                  END;
     3              STORE_TEXT ('MARKED_FOR_JOURNALING=');
     3              FAO_DESC[DSC$W_LENGTH] = MAX_FAO_LENGTH;
     3              FAO_DESC[DSC$A_POINTER] = FAO_BUF;
P    3              $FAO ($DESCRIPTOR ('!%D,'),
P    3                  FAO_DESC,
P    3                  FAO_DESC,
     3                  LOCAL_ACE[ACE$Q_ID_DATE]);
     3              FAO_BUF[11] = ':';
     3              IF .FAO_BUF[0] EQL ' '
     3              THEN
```

```
; 1222          1218 4                  BEGIN
; 1223          1219 4                  FAO_DESC[DSC$W_LENGTH] = .FAO_DESC[DSC$W_LENGTH] - 1;
; 1224          1220 4                  FAO_DESC[DSC$A_POINTER] = .FAO_DESC[DSC$A_POINTER] + 1;
; 1225          1221 3                  END;
; 1226          1222 3              CHECK_WIDTH (.FAO_DESC[DSC$W_LENGTH]);
; 1227          1223 3              CH$MOVE (.FAO_DESC[DSC$W_LENGTH], .FAO_DESC[DSC$A_POINTER], BUFFER[.SIZE]);
; 1228          1224 3              SIZE = .SIZE + .FAO_DESC[DSC$W_LENGTH];
; 1229          1225 2              END;
; 1230          1226 2
; 1231          1227 2          [INRANGE,
; 1232          1228 2           OUTRANGE]:
; 1233          1229 3              BEGIN
; 1234          1230 3              STORE_TEXT ('Unknown=');
; 1235          1231 3              CHECK_WIDTH (5);
; 1236          1232 3              FAO_DESCR[DSC$W_LENGTH] = 5;
; 1237          1233 3              FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
; 1238        P 1234 3              $FAOL (CTRSTR = $DESCRIPTOR ('%X!XB,'),
; 1239        P 1235 3                      OUTBUF = FAO_DESCR,
; 1240          1236 3                      PRMLST = %REF (.LOCAL_ACE[ACE$B_TYPE]));
; 1241          1237 3              SIZE = .SIZE + 5;
; 1242          1238 3              STORE_TEXT ('Size=');
; 1243          1239 3              CHECK_WIDTH (5);
; 1244          1240 3              FAO_DESCR[DSC$W_LENGTH] = 5;
; 1245          1241 3              FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
; 1246        P 1242 3              $FAOL (CTRSTR = $DESCRIPTOR ('%D!UB,'),
; 1247        P 1243 3                      OUTBUF = FAO_DESCR,
; 1248          1244 3                      PRMLST = %REF (.LOCAL_ACE[ACE$B_SIZE]));
; 1249          1245 3              SIZE = .SIZE + 5;
; 1250          1246 3              STORE_TEXT ('Flags=');
; 1251          1247 3              CHECK_WIDTH (7);
; 1252          1248 3              FAO_DESCR[DSC$W_LENGTH] = 7;
; 1253          1249 3              FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
; 1254        P 1250 3              $FAOL (CTRSTR = $DESCRIPTOR ('%X!XW,'),
; 1255        P 1251 3                      OUTBUF = FAO_DESCR,
; 1256          1252 3                      PRMLST = %REF (.LOCAL_ACE[ACE$W_FLAGS]));
; 1257          1253 3              SIZE = .SIZE + 7;
; 1258          1254 3              STORE_TEXT ('Access=');
; 1259          1255 3              CHECK_WIDTH (11);
; 1260          1256 3              FAO_DESCR[DSC$W_LENGTH] = 11;
; 1261          1257 3              FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
; 1262        P 1258 3              $FAOL (CTRSTR = $DESCRIPTOR ('%X!XL,'),
; 1263        P 1259 3                      OUTBUF = FAO_DESCR,
; 1264          1260 3                      PRMLST = %REF (.LOCAL_ACE[ACE$L_ACCESS]));
; 1265          1261 3              SIZE = .SIZE + 11;
; 1266          1262 3              STORE_TEXT ('Data=');
; 1267          1263 3              INCR J FROM 1 TO (.LOCAL_ACE[ACE$B_SIZE] - ACE$C_LENGTH + 3) / 4
; 1268          1264 3              DO
; 1269          1265 4                  BEGIN
; 1270          1266 4                  CHECK_WIDTH (11);
; 1271          1267 4                  FAO_DESCR[DSC$W_LENGTH] = 11;
; 1272          1268 4                  FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
; 1273        P 1269 4                  $FAOL (CTRSTR = $DESCRIPTOR ('%X!XL,'),
; 1274        P 1270 4                          OUTBUF = FAO_DESCR,
; 1275          1271 4                          PRMLST = VECTOR [LOCAL_ACE[ACE$L_KEY], .J - 1]);
; 1276          1272 4                  SIZE = .SIZE + 11;
; 1277          1273 3                  END;
; 1278          1274 3              BUFFER[.SIZE - 1] = %C')';
```

```
  3          IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_STRING)
  3          THEN
  4              BEGIN
  4              IF EXE$PROBEW (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
  4              THEN CH$COPY (.SIZE, BUFFER, 0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
  4              ELSE RETURN SS$_ACCVIO
  4              END
  3          ELSE RETURN SS$_ACCVIO;
  3          IF .ACL_LENGTH NEQ 0
  3          THEN IF PROBEW (%REF (0), %REF (4), .ACL_LENGTH)
  3               THEN .ACL_LENGTH = .SIZE ELSE RETURN SS$_ACCVIO;
  3          IF .SIZE GTR .ACL_STRING_LEN
  3          THEN RETURN SS$_BUFFEROVF ELSE RETURN SS$_NORMAL;
  2          END;
  2  TES;
  2
  2  ! Note any special flags applied to the ACE.
  2
  2  FLAGS = .LOCAL_ACE[ACE$W_FLAGS];
  2  IF .AUDIT_MASK
  2  THEN FLAGS = .FLAGS AND NOT ($FIELDMASK (ACE$V_SUCCESS) OR $FIELDMASK (ACE$V_FAILURE));
  2  IF .FLAGS NEQ 0
  2  THEN
  3      BEGIN
  3      STORE_TEXT ('OPTIONS=');
  3      IF TESTBITSC (FLAGS[$BITPOSITION (ACE$V_DEFAULT)])
  3      THEN STORE_TEXT ('DEFAULT+');
  3      IF TESTBITSC (FLAGS[$BITPOSITION (ACE$V_HIDDEN)])
  3      THEN STORE_TEXT ('HIDDEN+');
  3      IF TESTBITSC (FLAGS[$BITPOSITION (ACE$V_PROTECTED)])
  3      THEN STORE_TEXT ('PROTECTED+');
  3      IF TESTBITSC (FLAGS[$BITPOSITION (ACE$V_NOPROPAGATE)])
  3      THEN STORE_TEXT ('NOPROPAGATE+');
  3      IF .FLAGS NEQ 0
  3      THEN
  4          BEGIN
  4          CHECK_WIDTH (7);
  4          FAO_DESCR[DSC$W_LENGTH] = 7;
  4          FAO_DESCR[DSC$A_POINTER] = BUFFER[.SIZE];
P 4          $FAO ($DESCRIPTOR ('%X!XW,'),
P 4                FAO_DESCR,
P 4                FAO_DESCR,
  4                .FLAGS);
  4          SIZE = .SIZE + 7;
  4          END
  3      ELSE BUFFER[.SIZE - 1] = %C',';
  2      END;
  2
  2  ! Note the access rights.
  2
  2  IF .ACCESS_MASK
  2  THEN
  3      BEGIN
  3      IF .LOCAL_ACE[ACE$L_ACCESS] NEQ 0
  4      OR (.AUDIT_MASK
  5          AND (.LOCAL_ACE[ACE$V_SUCCESS]
  4               OR .LOCAL_ACE[ACE$V_FAILURE]))
```

```
3        THEN
4            BEGIN
4            STORE_TEXT ('ACCESS=');
4            INCR J FROM 0 TO 31
4            DO
5                BEGIN
5                IF .(LOCAL_ACE[ACE$L_ACCESS])<.J,1>
5                THEN
6                    BEGIN
6                    IF .BIT_NAME_TABLE NEQA 0
6                    THEN
7                        BEGIN
7                        IF PROBER (%REF (0), %REF (DSC$C_S_BLN), BIT_NAME_TABLE[.J, 0, 0, 0, 0])
7                        THEN BIT_NAME_DESC = BIT_NAME_TABLE[.J, 0, 0, 0, 0]
7                        ELSE RETURN SS$_ACCVIO;
7                        IF NOT EXE$PROBER (0, .BIT_NAME_DESC[DSC$W_LENGTH],
7                                              .BIT_NAME_DESC[DSC$A_POINTER])
7                        THEN RETURN SS$_ACCVIO;
7                        END
6                    ELSE BIT_NAME_DESC = .DEFAULT_BITS[.J];
6                    CHECK_WIDTH (.BIT_NAME_DESC[DSC$W_LENGTH] + 1);
6                    CH$MOVE (.BIT_NAME_DESC[DSC$W_LENGTH], .BIT_NAME_DESC[DSC$A_POINTER],
6                                                             BUFFER[.SIZE]);
6                    BUFFER[.SIZE + .BIT_NAME_DESC[DSC$W_LENGTH]] = '+';
6                    SIZE = .SIZE + .BIT_NAME_DESC[DSC$W_LENGTH] + 1;
5                    END;
4                END;
4            IF .AUDIT_MASK
4            THEN
5                BEGIN
5                IF .LOCAL_ACE[ACE$V_SUCCESS] THEN STORE_TEXT ('SUCCESS+');
5                IF .LOCAL_ACE[ACE$V_FAILURE] THEN STORE_TEXT ('FAILURE+');
4                END;
4            END
3        ELSE STORE_TEXT ('ACCESS=NONE+');
2        END;
2
2    ! Close off the ACE.
2
2    BUFFER[.SIZE - 1] = %C')';
2
2    ! Copy the formatted ACE to the user's buffer and return a size if required.
2
2    IF PROBER (%REF (0), %REF (DSC$C_S_BLN), .ACL_STRING)
2    THEN
3        BEGIN
3        IF EXE$PROBEW (0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
3        THEN CH$COPY (.SIZE, BUFFER, 0, .ACL_STRING_LEN, .ACL_STRING[DSC$A_POINTER])
3        ELSE RETURN SS$_ACCVIO
3        END
2    ELSE RETURN SS$_ACCVIO;
2    IF .ACL_LENGTH NEQ 0
2    THEN IF PROBEW (%REF (0), %REF (4), .ACL_LENGTH)
2         THEN .ACL_LENGTH = .SIZE ELSE RETURN SS$_ACCVIO;
2    IF .SIZE GTR .ACL_STRING_LEN
2    THEN RETURN SS$_BUFFEROVF ELSE RETURN SS$_NORMAL;
2
```

```
; 1393          1389  1 END;                                            ! End of routine SYS$FORMAT_ACL


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                         52  00324 P.ADF:  .ASCII  \R\                                ;
                                                         57  00325         .ASCII  \W\                                ;
                                                         45  00326         .ASCII  \E\                                ;
                                                         44  00327         .ASCII  \D\                                ;
                                                         43  00328         .ASCII  \C\                                ;
                                                        00#  00329         .BYTE   0[27]                              ;
                                             00  00  00  28  00344 P.ADG:  .ASCII  \(\<0><0><0>                       ;
             00  3D  52  45  49  46  49  54  4E  45  44  49  00348 P.ADH:  .ASCII  \IDENTIFIER=\<0>                   ;
                                                 49  25  21  00354 P.ADJ:  .ASCII  \!%I\                              ;
                                                             00357         .BLKB   1
                                                   00000003  00358 P.ADI:  .LONG   3                                  ;
                                                   00000000' 0035C         .ADDRESS P.ADJ                             ;
                                             00  00  00  2B  00360 P.ADK:  .ASCII  \+\<0><0><0>                       ;
             00  3D  4C  41  4E  52  55  4F  4A  5F  49  42  00364 P.ADL:  .ASCII  \BI_JOURNAL=\<0>                   ;
             00  3D  4C  41  4E  52  55  4F  4A  5F  49  41  00370 P.ADM:  .ASCII  \AI_JOURNAL=\<0>                   ;
             00  3D  4C  41  4E  52  55  4F  4A  5F  54  41  0037C P.ADN:  .ASCII  \AT_JOURNAL=\<0>                   ;
                                             00  00  00  2C  00388 P.ADO:  .ASCII  \,\<0><0><0>                       ;
 00  3D  4C  41  4E  52  55  4F  4A  5F  54  49  44  55  41  0038C P.ADP:  .ASCII  \AUDIT_JOURNAL=\<0><0>             ;
                                                         00  0039B                                                    ;
 00  3D  4C  41  4E  52  55  4F  4A  5F  4D  52  41  4C  41  0039C P.ADQ:  .ASCII  \ALARM_JOURNAL=\<0><0>             ;
                                                         00  003AB                                                    ;
                                             00  00  00  2C  003AC P.ADR:  .ASCII  \,\<0><0><0>                       ;
 54  43  45  54  4F  52  50  5F  54  4C  55  41  46  45  44  003B0 P.ADS:  .ASCII  \DEFAULT_PROTECTION,\<0>           ;
                                         00  2C  4E  4F  49  003BF                                                    ;
                             3A  4D  45  54  53  59  53      003C4 P.ADU:  .ASCII  \SYSTEM:\                          ;
                                                             003CB         .BLKB   1
                                                   00000007  003CC P.ADT:  .LONG   7                                  ;
                                                   00000000' 003D0         .ADDRESS P.ADU                             ;
                                 3A  52  45  4E  57  4F      003D4 P.ADW:  .ASCII  \OWNER:\                           ;
                                                             003DA         .BLKB   2
                                                   00000006  003DC P.ADV:  .LONG   6                                  ;
                                                   00000000' 003E0         .ADDRESS P.ADW                             ;
                                 3A  50  55  4F  52  47      003E4 P.ADY:  .ASCII  \GROUP:\                           ;
                                                             003EA         .BLKB   2
                                                   00000006  003EC P.ADX:  .LONG   6                                  ;
                                                   00000000' 003F0         .ADDRESS P.ADY                             ;
                                 3A  44  4C  52  4F  57      003F4 P.AEA:  .ASCII  \WORLD:\                           ;
                                                             003FA         .BLKB   2
                                                   00000006  003FC P.ADZ:  .LONG   6                                  ;
                                                   00000000' 00400         .ADDRESS P.AEA                             ;
                                             00  00  00  2C  00404 P.AEB:  .ASCII  \,\<0><0><0>                       ;
 2C  44  49  5F  4C  41  4E  52  55  4F  4A  5F  53  4D  52  00408 P.AEC:  .ASCII  \RMS_JOURNAL_ID,\<0>               ;
                                                         00  00417                                                    ;
                         00  00  00  24  4B  53  49  44      00418 P.AED:  .ASCII  \DISK$\<0><0><0>                   ;
 3D  45  4C  49  46  5F  44  45  4C  41  4E  52  55  4F  4A  00420 P.AEE:  .ASCII  \JOURNALED_FILE=\<0>               ;
                                                         00  0042F                                                    ;
                                             00  00  00  2C  00430 P.AEF:  .ASCII  \,\<0><0><0>                       ;
     2C  29  57  55  21  2C  57  55  21  2C  57  55  21  28  00434 P.AEH:  .ASCII  \(!UW,!UW,!UW),\                   ;
                                                             00442         .BLKB   2
                                                   0000000E  00444 P.AEG:  .LONG   14                                 ;
                                                   00000000' 00448         .ADDRESS P.AEH                             ;
```

```
52 55 4F 4A 5F 52 4F 46 5F 44 45 4B 52 41 4D 0044C P.AEI:  .ASCII  \MARKED_FOR_JOURNALING=\<0><0>                  ;
                  00 00 3D 47 4E 49 4C 41 4E 0045B                                                                 ;
                              2C 44 25 21 00464 P.AEK:  .ASCII  \!%D,\                                             ;
                                00000004 00468 P.AEJ:  .LONG   4                                                  ;
                                00000000' 0046C        .ADDRESS P.AEK                                             ;
                     3D 6E 77 6F 6E 6B 6E 55 00470 P.AEL:  .ASCII  \Unknown=\                                     ;
                           2C 42 58 21 58 25 00478 P.AEN:  .ASCII  \%X!XB,\                                       ;
                                           0047E        .BLKB   2
                                00000006 00480 P.AEM:  .LONG   6                                                  ;
                                00000000' 00484        .ADDRESS P.AEN                                             ;
                     00 00 00 3D 65 7A 69 53 00488 P.AEO:  .ASCII  \Size=\<0><0><0>                               ;
                           2C 42 55 21 44 25 00490 P.AEQ:  .ASCII  \%D!UB,\                                       ;
                                           00496        .BLKB   2
                                00000006 00498 P.AEP:  .LONG   6                                                  ;
                                00000000' 0049C        .ADDRESS P.AEQ                                             ;
                     00 00 3D 73 67 61 6C 46 004A0 P.AER:  .ASCII  \Flags=\<0><0>                                 ;
                           2C 57 58 21 58 25 004A8 P.AET:  .ASCII  \%X!XW,\                                       ;
                                           004AE        .BLKB   2
                                00000006 004B0 P.AES:  .LONG   6                                                  ;
                                00000000' 004B4        .ADDRESS P.AET                                             ;
                     00 3D 73 73 65 63 63 41 004B8 P.AEU:  .ASCII  \Access=\<0>                                   ;
                           2C 4C 58 21 58 25 004C0 P.AEW:  .ASCII  \%X!XL,\                                       ;
                                           004C6        .BLKB   2
                                00000006 004C8 P.AEV:  .LONG   6                                                  ;
                                00000000' 004CC        .ADDRESS P.AEW                                             ;
                     00 00 00 3D 61 74 61 44 004D0 P.AEX:  .ASCII  \Data=\<0><0><0>                               ;
                           2C 4C 58 21 58 25 004D8 P.AEZ:  .ASCII  \%X!XL,\                                       ;
                                           004DE        .BLKB   2
                                00000006 004E0 P.AEY:  .LONG   6                                                  ;
                                00000000' 004E4        .ADDRESS P.AEZ                                             ;
                     3D 53 4E 4F 49 54 50 4F 004E8 P.AFA:  .ASCII  \OPTIONS=\                                     ;
                     2B 54 4C 55 41 46 45 44 004F0 P.AFB:  .ASCII  \DEFAULT+\                                     ;
                     00 2B 4E 45 44 44 49 48 004F8 P.AFC:  .ASCII  \HIDDEN+\<0>                                   ;
         00 00 2B 44 45 54 43 45 54 4F 52 50 00500 P.AFD:  .ASCII  \PROTECTED+\<0><0>                             ;
         2B 45 54 41 47 41 50 4F 52 50 4F 4E 0050C P.AFE:  .ASCII  \NOPROPAGATE+\                                 ;
                           2C 57 58 21 58 25 00518 P.AFG:  .ASCII  \%X!XW,\                                       ;
                                           0051E        .BLKB   2
                                00000006 00520 P.AFF:  .LONG   6                                                  ;
                                00000000' 00524        .ADDRESS P.AFG                                             ;
                     00 3D 53 53 45 43 43 41 00528 P.AFH:  .ASCII  \ACCESS=\<0>                                   ;
                     2B 53 53 45 43 43 55 53 00530 P.AFI:  .ASCII  \SUCCESS+\                                     ;
                     2B 45 52 55 4C 49 41 46 00538 P.AFJ:  .ASCII  \FAILURE+\                                     ;
         2B 45 4E 4F 4E 3D 53 53 45 43 43 41 00540 P.AFK:  .ASCII  \ACCESS=NONE+\                                 ;

                                                 PROT_CODE=             P.ADF
                                                        .EXTRN  SYS$FAOL, SYS$FAO

                                                        .PSECT  $CODE$,NOWRT,2

                                     0FFC 00000         .ENTRY  SYS$FORMAT_ACL, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0820
                                                                R9,R10,R11                                     ;
                        5E      E87C  CE 9E 00002        MOVAB   -6020(SP), SP                                  ;
         0C  BC             08         00 0C 00007        PROBER  #0, #8, @ACL_STRING                            ; 0958
                                       79 13 0000C        BEQL    5$                                             ;
                        6E        0C  BC 3C 0000E        MOVZWL  @ACL_STRING, ACL_STRING_LEN                    ; 0961
              54       0C  AC         04 C1 00012        ADDL3   #4, ACL_STRING, R4                             ; 0962
                        50            64 D0 00017        MOVL    (R4), R0                                       ;
```

```
                                  51                6E  D0 0001A          MOVL    ACL_STRING_LEN, R1                        ;
                                                    53  D4 0001D          CLRL    R3                                        ;
                                       00000000G    00  16 0001F          JSB     EXE$PROBEW                                ;
                                  4F                50  E9 00025          BLBC    R0, 2$                                    ;
                   56       0C    AC                04  C1 00028          ADDL3   #4, ACL_STRING, R6                        ; 0963
        6E         20             6E                00  2C 0002D          MOVC5   #0, (SP), #32, ACL_STRING_LEN, @(R6)+     ;
                                                    96     00032                                                            ;
                                                    59  7C 00033          CLRQ    WIDTH                                     ; 0970
                                             08     AE  D4 00035          CLRL    TERM_POINTER                              ; 0971
                                                    5B  D4 00038          CLRL    TERM_LENGTH                               ;
                                             24     AE  94 0003A          CLRB    AUDIT_MASK                                ; 0972
                                             20     AE  94 0003D          CLRB    ACCESS_MASK                               ;
                                  50         10     AC  D0 00040          MOVL    LINE_WIDTH, R0                            ; 0976
                                                    09  13 00044          BEQL    1$                                        ;
                   60             04                00  0C 00046          PROBER  #0, #4, (R0)                              ; 0977
                                                    74  13 0004A          BEQL    11$                                       ;
                                  59                60  D0 0004C          MOVL    (R0), WIDTH                               ; 0978
                                             14     AC  D5 0004F  1$:     TSTL    TERM_DESC                                 ; 0981
                                                    29  13 00052          BEQL    4$                                        ;
               14  BC             08                00  0C 00054          PROBER  #0, #8, @TERM_DESC                        ; 0984
                                                    65  13 00059          BEQL    11$                                       ;
                                  5B         14     BC  3C 0005B          MOVZWL  @TERM_DESC, TERM_LENGTH                   ; 0987
                   50       14    AC                04  C1 0005F          ADDL3   #4, TERM_DESC, R0                         ; 0988
                            08    AE                60  D0 00064          MOVL    (R0), TERM_POINTER                        ;
                                  50         08     AE  D0 00068          MOVL    TERM_POINTER, R0                          ; 0989
                                  51                5B  D0 0006C          MOVL    TERM_LENGTH, R1                           ;
                                                    53  D4 0006F          CLRL    R3                                        ;
                                       00000000G    00  16 00071          JSB     EXE$PROBER                                ;
                                  03                50  E8 00077  2$:     BLBS    R0, 4$                                    ;
                                                  1087  31 0007A  3$:     BRW     172$                                      ;
                                  50         18     AC  D0 0007D  4$:     MOVL    LINE_INDENT, R0                           ; 0995
                                                    09  13 00081          BEQL    6$                                        ;
                   60             04                00  0C 00083          PROBER  #0, #4, (R0)                              ; 0996
                                                    F1  13 00087  5$:     BEQL    3$                                        ;
                                  5A                60  D0 00089          MOVL    (R0), INDENT                              ; 0997
                                                    5A  D5 0008C  6$:     TSTL    INDENT                                    ; 1000
                                                    1E  13 0008E          BEQL    10$                                       ;
                                                    59  D5 00090          TSTL    WIDTH                                     ; 1003
                                                    09  15 00092          BLEQ    7$                                        ;
                                  59                5A  D1 00094          CMPL    INDENT, WIDTH                             ; 1004
                                                    09  1B 00097          BLEQU   8$                                        ;
                                  50                14  D0 00099          MOVL    #20, R0                                   ;
                                                        04 0009C          RET                                               ;
                                  6E                5A  D1 0009D  7$:     CMPL    INDENT, ACL_STRING_LEN                    ; 1005
                                                    07  1A 000A0          BGTRU   9$                                        ;
                      00000C00    8F                5A  D1 000A2  8$:     CMPL    INDENT, #3072                             ; 1006
                                                    03  1B 000A9  9$:     BLEQU   10$                                       ;
                                                  105F  31 000AB          BRW     174$                                      ;
                                       00000000'    EF  D4 000AE  10$:    CLRL    BIT_NAME_TABLE                            ; 1013
                                  50         1C     AC  D0 000B4          MOVL    BIT_TABLE, R0                             ; 1014
                                                    0F  13 000B8          BEQL    12$                                       ;
                   60     0100    8F                00  0C 000BA          PROBER  #0, #256, (R0)                            ; 1015
                                                    B8  13 000C0  11$:    BEQL    3$                                        ;
                      00000000'   EF                50  D0 000C2          MOVL    R0, BIT_NAME_TABLE                        ; 1016
        5A         20             6E                00  2C 000C9  12$:    MOVC5   #0, (SP), #32, INDENT, BUFFER             ; 1021
                                           0A5C     CE     000CE                                                            ;
                                  57                5A  D0 000D1          MOVL    INDENT, LINE_SIZE                         ; 1022
```

```
                          56             5A  D0 000D4          MOVL    INDENT, SIZE                                    :
                                  28     AE  D4 000D7          CLRL    40(SP)                                          : 1023
                                         59  D5 000DA          TSTL    WIDTH                                           :
                                         2E  13 000DC          BEQL    14$                                             :
                                  28     AE  D6 000DE          INCL    40(SP)                                          :
                          50      01     A7  9E 000E1          MOVAB   1(R7), R0                                       :
                          59             50  D1 000E5          CMPL    R0, WIDTH                                       :
                                         22  1B 000E8          BLEQU   14$                                             :
                                         5B  D5 000EA          TSTL    TERM_LENGTH                                     :
                                         0F  15 000EC          BLEQ    13$                                             :
          0A5C CE46       08     BE      5B  28 000EE          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]        :
                          50      14     BC  3C 000F6          MOVZWL  @TERM_DESC, R0                                  :
                          56             50  C0 000FA          ADDL2   R0, SIZE                                        :
    5A            20      6E             00  2C 000FD 13$:     MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]             :
                             0A5C CE46          00102                                                                  :
                          56             5A  C0 00106          ADDL2   INDENT, SIZE                                    :
                          57             5A  D0 00109          MOVL    INDENT, LINE_SIZE                               :
                                         57  D6 0010C 14$:     INCL    LINE_SIZE                                       :
               0A5C CE46 00000000'       EF  90 0010E          MOVB    P.ADG, BUFFER[SIZE]                             :
                                         56  D6 00118          INCL    SIZE                                            :
                          54      04     AC  D0 0011A          MOVL    ACL_ENTRY, R4                                   : 1024
                  64      08             00  0C 0011E          PROBER  #0, #8, (R4)                                    :
                                         24  13 00122          BEQL    16$                                             :
                          55             64  3C 00124          MOVZWL  (R4), ACL_ENTRY_LEN                             : 1027
                 000000FF        8F      55  D1 00127          CMPL    ACL_ENTRY_LEN, #255                             : 1028
                                         06  1B 0012E          BLEQU   15$                                             :
                          50    21E4     8F  3C 00130          MOVZWL  #8676, R0                                       :
                                             04 00135          RET                                                     :
                          50      04     A4  D0 00136 15$:     MOVL    4(R4), R0                                       : 1029
                          51             55  D0 0013A          MOVL    ACL_ENTRY_LEN, R1                               :
                                         53  D4 0013D          CLRL    R3                                              :
                         00000000G       00  16 0013F          JSB     EXE$PROBER                                      :
                          03             50  E8 00145          BLBS    R0, 17$                                         :
                                   0FB9      31 00148 16$:     BRW     172$                                            :
          FF00    CD      04     B4      55  28 0014B 17$:     MOVC3   ACL_ENTRY_LEN, @4(R4), LOCAL_ACE                : 1030
                          18     AE    FF01  CD  9A 00152      MOVZBL  LOCAL_ACE+1, 24(SP)                             : 1037
                  08      01      18     AE  8F 00158          CASEB   24(SP), #1, #8                                  :
    04A8          04A8    04A8         0387     0015D 18$:     .WORD   46$-18$,-                                       :
    08A4          0012    0609         0609     00165                  55$-18$,-                                       :
                                       072E     0016D                  55$-18$,-                                       :
                                                                       55$-18$,-                                       :
                                                                       69$-18$,-                                       :
                                                                       69$-18$,-                                       :
                                                                       19$-18$,-                                       :
                                                                       98$-18$,-                                       :
                                                                       81$-18$                                         :
                          2B      28     AE  E9 0016F 19$:     BLBC    40(SP), 21$                                     : 1230
                          50      08     A7  9E 00173          MOVAB   8(R7), R0                                       :
                          59             50  D1 00177          CMPL    R0, WIDTH                                       :
                                         22  1B 0017A          BLEQU   21$                                             :
                                         5B  D5 0017C          TSTL    TERM_LENGTH                                     :
                                         0F  15 0017E          BLEQ    20$                                             :
          0A5C CE46       08     BE      5B  28 00180          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]        :
                          50      14     BC  3C 00188          MOVZWL  @TERM_DESC, R0                                  :
                          56             50  C0 0018C          ADDL2   R0, SIZE                                        :
    5A            20      6E             00  2C 0018F 20$:     MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]             :
                             0A5C CE46          00194                                                                  :
                               56           5A  C0 00198        ADDL2   INDENT, SIZE                                ;
                               57           5A  D0 0019B        MOVL    INDENT, LINE_SIZE                           ;
                               57           08  C0 0019E 21$:   ADDL2   #8, LINE_SIZE                               ;
        0A5C CE46 00000000'    EF           08  28 001A1        MOVC3   #8, P.AEC, BUFFER[SIZE]                     ;
                               56           08  C0 001AC        ADDL2   #8, SIZE                                    ;
                               2B      28   AE  E9 001AF        BLBC    40(SP), 23$                                 ; 1231
                               50      05   A7  9E 001B3        MOVAB   5(R7), R0                                   ;
                               59           50  D1 001B7        CMPL    R0, WIDTH                                   ;
                                            22  1B 001BA        BLEQU   23$                                         ;
                                            5B  D5 001BC        TSTL    TERM_LENGTH                                 ;
                                            0F  15 001BE        BLEQ    22$                                         ;
        0A5C CE46        08    BE           5B  28 001C0        MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    ;
                               50      14   BC  3C 001C8        MOVZWL  @TERM_DESC, R0                              ;
                               56           50  C0 001CC        ADDL2   R0, SIZE                                    ;
  5A             20            6E           00  2C 001CF 22$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         ;
                                   0A5C CE46       001D4                                                        ;
                               56           5A  C0 001D8        ADDL2   INDENT, SIZE                                ;
                               57           5A  D0 001DB        MOVL    INDENT, LINE_SIZE                           ;
                               57           05  C0 001DE 23$:   ADDL2   #5, LINE_SIZE                               ;
                       FEF8    CD           05  B0 001E1        MOVW    #5, FAO_DESCR                               ; 1232
                       FEFC    CD  0A5C CE46    9E 001E6        MOVAB   BUFFER[SIZE], FAO_DESCR+4                   ; 1233
                         1C    AE      18   AE  D0 001EE        MOVL    24(SP), 28(SP)                              ; 1236
                                       1C   AE  9F 001F3        PUSHAB  28(SP)                                      ;
                                     FEF8   CD  9F 001F6        PUSHAB  FAO_DESCR                                   ;
                                            7E  D4 001FA        CLRL    -(SP)                                       ;
                                00000000'   EF  9F 001FC        PUSHAB  P.AEM                                       ;
                  00000000G    00           04  FB 00202        CALLS   #4, SYS$FAOL                                ;
                               56           05  C0 00209        ADDL2   #5, SIZE                                    ; 1237
                               2B      28   AE  E9 0020C        BLBC    40(SP), 25$                                 ; 1238
                               50      05   A7  9E 00210        MOVAB   5(R7), R0                                   ;
                               59           50  D1 00214        CMPL    R0, WIDTH                                   ;
                                            22  1B 00217        BLEQU   25$                                         ;
                                            5B  D5 00219        TSTL    TERM_LENGTH                                 ;
                                            0F  15 0021B        BLEQ    24$                                         ;
        0A5C CE46        08    BE           5B  28 0021D        MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    ;
                               50      14   BC  3C 00225        MOVZWL  @TERM_DESC, R0                              ;
                               56           50  C0 00229        ADDL2   R0, SIZE                                    ;
  5A             20            6E           00  2C 0022C 24$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         ;
                                   0A5C CE46       00231                                                        ;
                               56           5A  C0 00235        ADDL2   INDENT, SIZE                                ;
                               57           5A  D0 00238        MOVL    INDENT, LINE_SIZE                           ;
                               57           05  C0 0023B 25$:   ADDL2   #5, LINE_SIZE                               ;
        0A5C CE46 00000000'    EF           05  28 0023E        MOVC3   #5, P.AEO, BUFFER[SIZE]                     ;
                               56           05  C0 00249        ADDL2   #5, SIZE                                    ;
                               2B      28   AE  E9 0024C        BLBC    40(SP), 27$                                 ; 1239
                               50      05   A7  9E 00250        MOVAB   5(R7), R0                                   ;
                               59           50  D1 00254        CMPL    R0, WIDTH                                   ;
                                            22  1B 00257        BLEQU   27$                                         ;
                                            5B  D5 00259        TSTL    TERM_LENGTH                                 ;
                                            0F  15 0025B        BLEQ    26$                                         ;
        0A5C CE46        08    BE           5B  28 0025D        MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    ;
                               50      14   BC  3C 00265        MOVZWL  @TERM_DESC, R0                              ;
                               56           50  C0 00269        ADDL2   R0, SIZE                                    ;
  5A             20            6E           00  2C 0026C 26$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         ;
                                   0A5C CE46       00271                                                        ;
                               56           5A  C0 00275        ADDL2   INDENT, SIZE                                ;
                               57           5A  D0 00278        MOVL    INDENT, LINE_SIZE                           ;
```

```
                        57                 05  C0 0027B 27$:    ADDL2   #5, LINE_SIZE                                  :
                  FEF8  CD                 05  B0 0027E         MOVW    #5, FAO_DESCR                                  : 1240
                  FEFC  CD      0A5C CE46  9E 00283             MOVAB   BUFFER[SIZE], FAO_DESCR+4                      : 1241
                    1C  AE           FF00  CD  9A 0028B         MOVZBL  LOCAL_ACE, 28(SP)                              : 1244
                                       1C  AE  9F 00291         PUSHAB  28(SP)                                         :
                                     FEF8  CD  9F 00294         PUSHAB  FAO_DESCR                                      :
                                           7E  D4 00298         CLRL    -(SP)                                          :
                                00000000'  EF  9F 0029A         PUSHAB  P.AEP                                          :
                  00000000G  00            04  FB 002A0         CALLS   #4, SYS$FAOL                                   :
                        56                 05  C0 002A7         ADDL2   #5, SIZE                                       : 1245
                        2B       28        AE  E9 002AA         BLBC    40(SP), 29$                                    : 1246
                        50       06        A7  9E 002AE         MOVAB   6(R7), R0                                      :
                        59                 50  D1 002B2         CMPL    R0, WIDTH                                      :
                                           22  1B 002B5         BLEQU   29$                                            :
                                           5B  D5 002B7         TSTL    TERM_LENGTH                                    :
                                           0F  15 002B9         BLEQ    28$                                            :
       0A5C CE46       08       BE         5B  28 002BB         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]       :
                        50       14        BC  3C 002C3         MOVZWL  @TERM_DESC, R0                                 :
                        56                 50  C0 002C7         ADDL2   R0, SIZE                                       :
   5A          20       6E                 00  2C 002CA 28$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]            :
                                0A5C CE46     002CF                                                                    :
                        56                 5A  C0 002D3         ADDL2   INDENT, SIZE                                   :
                        57                 5A  D0 002D6         MOVL    INDENT, LINE_SIZE                              :
                        57                 06  C0 002D9 29$:    ADDL2   #6, LINE_SIZE                                  :
       0A5C CE46 00000000'  EF             06  28 002DC         MOVC3   #6, P.AER, BUFFER[SIZE]                        :
                        56                 06  C0 002E7         ADDL2   #6, SIZE                                       :
                        2B       28        AE  E9 002EA         BLBC    40(SP), 31$                                    : 1247
                        50       07        A7  9E 002EE         MOVAB   7(R7), R0                                      :
                        59                 50  D1 002F2         CMPL    R0, WIDTH                                      :
                                           22  1B 002F5         BLEQU   31$                                            :
                                           5B  D5 002F7         TSTL    TERM_LENGTH                                    :
                                           0F  15 002F9         BLEQ    30$                                            :
       0A5C CE46       08       BE         5B  28 002FB         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]       :
                        50       14        BC  3C 00303         MOVZWL  @TERM_DESC, R0                                 :
                        56                 50  C0 00307         ADDL2   R0, SIZE                                       :
   5A          20       6E                 00  2C 0030A 30$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]            :
                                0A5C CE46     0030F                                                                    :
                        56                 5A  C0 00313         ADDL2   INDENT, SIZE                                   :
                        57                 5A  D0 00316         MOVL    INDENT, LINE_SIZE                              :
                        57                 07  C0 00319 31$:    ADDL2   #7, LINE_SIZE                                  :
                  FEF8  CD                 07  B0 0031C         MOVW    #7, FAO_DESCR                                  : 1248
                  FEFC  CD      0A5C CE46  9E 00321             MOVAB   BUFFER[SIZE], FAO_DESCR+4                      : 1249
                    1C  AE           FF02  CD  3C 00329         MOVZWL  LOCAL_ACE+2, 28(SP)                            : 1252
                                       1C  AE  9F 0032F         PUSHAB  28(SP)                                         :
                                     FEF8  CD  9F 00332         PUSHAB  FAO_DESCR                                      :
                                           7E  D4 00336         CLRL    -(SP)                                          :
                                00000000'  EF  9F 00338         PUSHAB  P.AES                                          :
                  00000000G  00            04  FB 0033E         CALLS   #4, SYS$FAOL                                   :
                        56                 07  C0 00345         ADDL2   #7, SIZE                                       : 1253
                        2B       28        AE  E9 00348         BLBC    40(SP), 33$                                    : 1254
                        50       07        A7  9E 0034C         MOVAB   7(R7), R0                                      :
                        59                 50  D1 00350         CMPL    R0, WIDTH                                      :
                                           22  1B 00353         BLEQU   33$                                            :
                                           5B  D5 00355         TSTL    TERM_LENGTH                                    :
                                           0F  15 00357         BLEQ    32$                                            :
       0A5C CE46       08       BE         5B  28 00359         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]       :
                        50       14        BC  3C 00361         MOVZWL  @TERM_DESC, R0                                 :
```

```
                                        56             50  C0 00365            ADDL2   R0, SIZE                                    :
        5A             20               6E             00  2C 00368 32$:       MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         :
                                                 0A5C CE46    0036D                                                                :
                                        56             5A  C0 00371            ADDL2   INDENT, SIZE                                :
                                        57             5A  D0 00374            MOVL    INDENT, LINE_SIZE                           :
                                        57             07  C0 00377 33$:       ADDL2   #7, LINE_SIZE                               :
               0A5C CE46 00000000'      EF             07  28 0037A            MOVC3   #7, P.AEU, BUFFER[SIZE]                     :
                                        56             07  C0 00385            ADDL2   #7, SIZE                                    :
                                        2B        28   AE  E9 00388            BLBC    40(SP), 35$                                 :  1255
                                        50        0B   A7  9E 0038C            MOVAB   11(R7), R0                                  :
                                        59             50  D1 00390            CMPL    R0, WIDTH                                   :
                                                       22  1B 00393            BLEQU   35$                                         :
                                                       5B  D5 00395            TSTL    TERM_LENGTH                                 :
                                                       0F  15 00397            BLEQ    34$                                         :
               0A5C CE46        08      BE             5B  28 00399            MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    :
                                        50        14   BC  3C 003A1            MOVZWL  @TERM_DESC, R0                              :
                                        56             50  C0 003A5            ADDL2   R0, SIZE                                    :
        5A             20               6E             00  2C 003A8 34$:       MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         :
                                                 0A5C CE46    003AD                                                                :
                                        56             5A  C0 003B1            ADDL2   INDENT, SIZE                                :
                                        57             5A  D0 003B4            MOVL    INDENT, LINE_SIZE                           :
                                        57             0B  C0 003B7 35$:       ADDL2   #11, LINE_SIZE                              :
                              FEF8      CD             0B  B0 003BA            MOVW    #11, FAO_DESCR                              :  1256
                              FEFC      CD       0A5C CE46 9E 003BF            MOVAB   BUFFER[SIZE], FAO_DESCR+4                   :  1257
                                1C      AE          FF04   CD D0 003C7         MOVL    LOCAL_ACE+4, 28(SP)                         :  1260
                                                    1C   AE  9F 003CD          PUSHAB  28(SP)                                      :
                                                  FEF8   CD  9F 003D0          PUSHAB  FAO_DESCR                                   :
                                                         7E  D4 003D4          CLRL    -(SP)                                       :
                                             00000000'   EF  9F 003D6          PUSHAB  P.AEV                                       :
                         00000000G      00             04  FB 003DC            CALLS   #4, SYS$FAOL                                :
                                        56             0B  C0 003E3            ADDL2   #11, SIZE                                   :  1261
                                        2B        28   AE  E9 003E6            BLBC    40(SP), 37$                                 :  1262
                                        50        05   A7  9E 003EA            MOVAB   5(R7), R0                                   :
                                        59             50  D1 003EE            CMPL    R0, WIDTH                                   :
                                                       22  1B 003F1            BLEQU   37$                                         :
                                                       5B  D5 003F3            TSTL    TERM_LENGTH                                 :
                                                       0F  15 003F5            BLEQ    36$                                         :
               0A5C CE46        08      BE             5B  28 003F7            MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    :
                                        50        14   BC  3C 003FF            MOVZWL  @TERM_DESC, R0                              :
                                        56             50  C0 00403            ADDL2   R0, SIZE                                    :
        5A             20               6E             00  2C 00406 36$:       MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         :
                                                 0A5C CE46    0040B                                                                :
                                        56             5A  C0 0040F            ADDL2   INDENT, SIZE                                :
                                        57             5A  D0 00412            MOVL    INDENT, LINE_SIZE                           :
                                        57             05  C0 00415 37$:       ADDL2   #5, LINE_SIZE                               :
               0A5C CE46 00000000'      EF             05  28 00418            MOVC3   #5, P.AEX, BUFFER[SIZE]                     :
                                        56             05  C0 00423            ADDL2   #5, SIZE                                    :
                                1C      AE        FF00   CD  9A 00426          MOVZBL  LOCAL_ACE, 28(SP)                           :  1263
                                1C      AE             05  C2 0042C            SUBL2   #5, 28(SP)                                  :
                                1C      AE             04  C6 00430            DIVL2   #4, 28(SP)                                  :
                                                       58  D4 00434            CLRL    J                                           :
                                                       5A  11 00436            BRB     41$                                         :
                                        2B        28   AE  E9 00438 38$:       BLBC    40(SP), 40$                                 :  1266
                                        50        0B   A7  9E 0043C            MOVAB   11(R7), R0                                  :
                                        59             50  D1 00440            CMPL    R0, WIDTH                                   :
                                                       22  1B 00443            BLEQU   40$                                         :
                                                       5B  D5 00445            TSTL    TERM_LENGTH                                 :
```

```
                                       0F  15 00447           BLEQ    39$                                                 :
       0A5C CE46      08   BE          5B  28 00449           MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]            :
                           50      14  BC  3C 00451           MOVZWL  @TERM_DESC, R0                                      :
                           56          50  C0 00455           ADDL2   R0, SIZE                                            :
  5A             20        6E          00  2C 00458 39$:      MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]                 :
                                  0A5C CE46   0045D                                                                       :
                           56          5A  C0 00461           ADDL2   INDENT, SIZE                                        :
                           57          5A  D0 00464           MOVL    INDENT, LINE_SIZE                                   :
                           57          0B  C0 00467 40$:      ADDL2   #11, LINE_SIZE                                      :
                     FEF8  CD          0B  B0 0046A           MOVW    #11, FAO_DESCR                                      : 1267
                     FEFC  CD     0A5C CE46 9E 0046F          MOVAB   BUFFER[SIZE], FAO_DESCR+4                           : 1268
                                  FF04 CD48 DF 00477          PUSHAL  LOCAL_ACE+4[J]                                      : 1271
                                       FEF8 CD 9F 0047C       PUSHAB  FAO_DESCR                                           :
                                          7E  D4 00480        CLRL    -(SP)                                               :
                               00000000' EF  9F 00482         PUSHAB  P.AEY                                               :
                 00000000G  00          04  FB 00488          CALLS   #4, SYS$FAOL                                        :
                           56          0B  C0 0048F           ADDL2   #11, SIZE                                           : 1272
                 A1        58      1C  AE  F3 00492 41$:      AOBLEQ  28(SP), J, 38$                                      : 1263
                 0A5B CE46          29  90 00497              MOVB    #41, BUFFER-1[SIZE]                                 : 1274
          0C     BC        08          00  0C 0049D           PROBER  #0, #8, @ACL_STRING                                 : 1275
                                       16  13 004A2           BEQL    42$                                                 :
                 54        0C   AC     04  C1 004A4           ADDL3   #4, ACL_STRING, R4                                  : 1278
                           50          64  D0 004A9           MOVL    (R4), R0                                            :
                           51          6E  D0 004AC           MOVL    ACL_STRING_LEN, R1                                  :
                                       53  D4 004AF           CLRL    R3                                                  :
                               00000000G  00  16 004B1        JSB     EXE$PROBEW                                          :
                           03          50  E8 004B7           BLBS    R0, 43$                                             :
                                     0C47  31 004BA 42$:      BRW     172$                                                :
                 58        0C   AC     04  C1 004BD 43$:      ADDL3   #4, ACL_STRING, R8                                  : 1279
  6E             00      0A5C   CE     56  2C 004C2           MOVC5   SIZE, BUFFER, #0, ACL_STRING_LEN, @(R8)+            :
                                       98     004C9                                                                       :
                           50      08  AC  D0 004CA           MOVL    ACL_LENGTH, R0                                      : 1283
                                       09  13 004CE           BEQL    44$                                                 :
                 60        04          00  0D 004D0           PROBEW  #0, #4, (R0)                                        : 1284
                                       E4  13 004D4           BEQL    42$                                                 :
                           60          56  D0 004D6           MOVL    SIZE, (R0)                                          : 1285
                           6E          56  D1 004D9 44$:      CMPL    SIZE, ACL_STRING_LEN                                : 1286
                                       03  14 004DC           BGTR    45$                                                 :
                                     0C32  31 004DE           BRW     175$                                                :
                                     0C29  31 004E1 45$:      BRW     174$                                                :
                 20        AE          01  90 004E4 46$:      MOVB    #1, ACCESS_MASK                                     : 1041
                           2B      28  AE  E9 004E8           BLBC    40(SP), 48$                                         : 1042
                           50      0B  A7  9E 004EC           MOVAB   11(R7), R0                                          :
                           59          50  D1 004F0           CMPL    R0, WIDTH                                           :
                                       22  1B 004F3           BLEQU   48$                                                 :
                                       5B  D5 004F5           TSTL    TERM_LENGTH                                         :
                                       0F  15 004F7           BLEQ    47$                                                 :
       0A5C CE46      08   BE          5B  28 004F9           MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]            :
                           50      14  BC  3C 00501           MOVZWL  @TERM_DESC, R0                                      :
                           56          50  C0 00505           ADDL2   R0, SIZE                                            :
  5A             20        6E          00  2C 00508 47$:      MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]                 :
                                  0A5C CE46   0050D                                                                       :
                           56          5A  C0 00511           ADDL2   INDENT, SIZE                                        :
                           57          5A  D0 00514           MOVL    INDENT, LINE_SIZE                                   :
                           57          0B  C0 00517 48$:      ADDL2   #11, LINE_SIZE                                      :
       0A5C CE46 00000000'  EF         0B  28 0051A           MOVC3   #11, P.ADA, BUFFER[SIZE]                            :
                           56          0B  C0 00525           ADDL2   #11, SIZE                                           :
```

```
                         1C   AE      FF00   CD  9A 00528         MOVZBL  LOCAL_ACE, 28(SP)                               ; 1043
                         1C   AE        05   C2  0052E            SUBL2   #5, 28(SP)
                         1C   AE        04   C6  00532            DIVL2   #4, 28(SP)
    58    FF02   CD           04             00  EF 00536         EXTZV   #0, #4, LOCAL_ACE+2, J
                                      00B5   31  0053D            BRW     54$
                         2C   AE      FF04 CD48  D0 00540  49$:   MOVL    LOCAL_ACE+4[J], KEY_IDENTIFIER                  ; 1046
                       0A54   CE      0200   8F  B0 00547         MOVW    #512, FAO_DESC                                  ; 1047
                       0A58   CE      0854   CE  9E 0054E         MOVAB   FAO_BUF, FAO_DESC+4                             ; 1048
                                        2C   AE  9F 00555         PUSHAB  KEY_IDENTIFIER                                  ; 1052
                                      0A58   CE  9F 00558         PUSHAB  FAO_DESC
                                      0A5C   CE  9F 0055C         PUSHAB  FAO_DESC
                                  00000000'  EF  9F 00560         PUSHAB  P.ADI
                     00000000G   00           04  FB 00566         CALLS   #4, SYS$FAOL
                                 2F     28   AE  E9 0056D         BLBC    40(SP), 51$                                     ; 1053
                                 50   0A54   CE  3C 00571         MOVZWL  FAO_DESC, R0
                                 50          57  C0 00576         ADDL2   LINE_SIZE, R0
                                 59          50  D1 00579         CMPL    R0, WIDTH
                                             22  1B 0057C         BLEQU   51$
                                             5B  D5 0057E         TSTL    TERM_LENGTH
                                             0F  15 00580         BLEQ    50$
          0A5C CE46        08    BE          5B  28 00582         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]
                                 50     14   BC  3C 0058A         MOVZWL  @TERM_DESC, R0
                                 56          50  C0 0058E         ADDL2   R0, SIZE
    5A             20            6E          00  2C 00591  50$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]
                                    0A5C CE46       00596
                                 56          5A  C0 0059A         ADDL2   INDENT, SIZE
                                 57          5A  D0 0059D         MOVL    INDENT, LINE_SIZE
                         18      AE   0A54   CE  3C 005A0  51$:   MOVZWL  FAO_DESC, 24(SP)
                                 57     18   AE  C0 005A6         ADDL2   24(SP), LINE_SIZE
          0A5C CE46      0A58    DE     18   AE  28 005AA         MOVC3   24(SP), @FAO_DESC+4, BUFFER[SIZE]               ; 1056
                                 56     18   AE  C0 005B4         ADDL2   24(SP), SIZE                                    ; 1057
                                 2B     28   AE  E9 005B8         BLBC    40(SP), 53$                                     ; 1058
                                 50     01   A7  9E 005BC         MOVAB   1(R7), R0
                                 59          50  D1 005C0         CMPL    R0, WIDTH
                                             22  1B 005C3         BLEQU   53$
                                             5B  D5 005C5         TSTL    TERM_LENGTH
                                             0F  15 005C7         BLEQ    52$
          0A5C CE46        08    BE          5B  28 005C9         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]
                                 50     14   BC  3C 005D1         MOVZWL  @TERM_DESC, R0
                                 56          50  C0 005D5         ADDL2   R0, SIZE
    5A             20            6E          00  2C 005D8  52$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]
                                    0A5C CE46       005DD
                                 56          5A  C0 005E1         ADDL2   INDENT, SIZE
                                 57          5A  D0 005E4         MOVL    INDENT, LINE_SIZE
                                             57  D6 005E7  53$:   INCL    LINE_SIZE
                     0A5C CE46 00000000'     EF  90 005E9         MOVB    P.ADR, BUFFER[SIZE]
                                             56  D6 005F3         INCL    SIZE
  FF44             58            01     1C   AE  F1 005F5  54$:   ACBL    28(SP), #1, J, 49$                              ; 1043
                     0A5B CE46               2C  90 005FC         MOVB    #44, BUFFER-1[SIZE]                             ; 1060
                                      0718   31  00602            BRW     126$                                            ; 1037
                                 02     18   AE  91 00605  55$:   CMPB    24(SP), #2                                      ; 1066
                                             40  12 00609         BNEQ    58$
                                 2B     28   AE  E9 0060B         BLBC    40(SP), 57$                                     ; 1067
                                 50     0B   A7  9E 0060F         MOVAB   11(R7), R0
                                 59          50  D1 00613         CMPL    R0, WIDTH
                                             22  1B 00616         BLEQU   57$
                                             5B  D5 00618         TSTL    TERM_LENGTH                                     ;
                                      0F  15 0061A        BLEQ     56$                                          :
        0A5C CE46       08     BE     5B  28 0061C        MOVC3    TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]     :
                               50     14  BC  3C 00624    MOVZWL   @TERM_DESC, R0                               :
                               56     50  C0 00628        ADDL2    R0, SIZE                                     :
   5A          20              6E     00  2C 0062B 56$:   MOVC5    #0, (SP), #32, INDENT, BUFFER[SIZE]          :
                                0A5C CE46    00630                                                          :
                               56     5A  C0 00634        ADDL2    INDENT, SIZE                                 :
                               57     5A  D0 00637        MOVL     INDENT, LINE_SIZE                            :
                               57     0B  C0 0063A 57$:   ADDL2    #11, LINE_SIZE                               :
        0A5C CE46 00000000'    EF     0B  28 0063D        MOVC3    #11, P.AD[, BUFFER[SIZE]                     :
                               56     0B  C0 00648        ADDL2    #11, SIZE                                    :
                               03     18  AE  91 0064B 58$:   CMPB     24(SP), #3                               : 1068
                                      40  12 0064F        BLEQ     61$                                          :
                               2B     28  AE  E9 00651    BLBC     40(SP), 60$                                  : 1069
                               50     0B  A7  9E 00655    MOVAB    11(R7), R0                                   :
                               59     50  D1 00659        CMPL     R0, WIDTH                                    :
                                      22  1B 0065C        BLEQU    60$                                          :
                                      5B  D5 0065E        TSTL     TERM_LENGTH                                  :
                                      0F  15 00660        BLEQ     59$                                          :
        0A5C CE46       08     BE     5B  28 00662        MOVC3    TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]     :
                               50     14  BC  3C 0066A    MOVZWL   @TERM_DESC, R0                               :
                               56     50  C0 0066E        ADDL2    R0, SIZE                                     :
   5A          20              6E     00  2C 00671 59$:   MOVC5    #0, (SP), #32, INDENT, BUFFER[SIZE]          :
                                0A5C CE46    00676                                                          :
                               56     5A  C0 0067A        ADDL2    INDENT, SIZE                                 :
                               57     5A  D0 0067D        MOVL     INDENT, LINE_SIZE                            :
                               57     0B  C0 00680 60$:   ADDL2    #11, LINE_SIZE                               :
        0A5C CE46 00000000'    EF     0B  28 00683        MOVC3    #11, P.ADM, BUFFER[SIZE]                     :
                               56     0B  C0 0068E        ADDL2    #11, SIZE                                    :
                               04     18  AE  91 00691 61$:   CMPB     24(SP), #4                               : 1070
                                      40  12 00695        BNEQ     64$                                          :
                               2B     28  AE  E9 00697    BLBC     40(SP), 63$                                  : 1071
                               50     0B  A7  9E 0069B    MOVAB    11(R7), R0                                   :
                               59     50  D1 0069F        CMPL     R0, WIDTH                                    :
                                      22  1B 006A2        BLEQU    63$                                          :
                                      5B  D5 006A4        TSTL     TERM_LENGTH                                  :
                                      0F  15 006A6        BLEQ     62$                                          :
        0A5C CE46       08     BE     5B  28 006A8        MOVC3    TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]     :
                               50     14  BC  3C 006B0    MOVZWL   @TERM_DESC, R0                               :
                               56     50  C0 006B4        ADDL2    R0, SIZE                                     :
   5A          20              6E     00  2C 006B7 62$:   MOVC5    #0, (SP), #32, INDENT, BUFFER[SIZE]          :
                                0A5C CE46    006BC                                                          :
                               56     5A  C0 006C0        ADDL2    INDENT, SIZE                                 :
                               57     5A  D0 006C3        MOVL     INDENT, LINE_SIZE                            :
                               57     0B  C0 006C6 63$:   ADDL2    #11, LINE_SIZE                               :
        0A5C CE46 00000000'    EF     0B  28 006C9        MOVC3    #11, P.ADN, BUFFER[SIZE]                     :
                               56     0B  C0 006D4        ADDL2    #11, SIZE                                    :
                               31     28  AE  E9 006D7 64$:   BLBC     40(SP), 66$                              : 1072
                               50   FF00  CD  9A 006DB    MOVZBL   LOCAL_ACE, R0                                :
                               50   FC A047   9E 006E0    MOVAB    -4(R0)[LINE_SIZE], R0                        :
                               59     50  D1 006E5        CMPL     R0, WIDTH                                    :
                                      22  1B 006E8        BLEQU    66$                                          :
                                      5B  D5 006EA        TSTL     TERM_LENGTH                                  :
                                      0F  15 006EC        BLEQ     65$                                          :
        0A5C CE46       08     BE     5B  28 006EE        MOVC3    TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]     :
                               50     14  BC  3C 006F6    MOVZWL   @TERM_DESC, R0                               :
                               56     50  C0 006FA        ADDL2    R0, SIZE                                     :
```

```
           5A            20            6E            00  2C 006FD 65$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]       ;
                                              0A5C CE46     00702                                                            ;
                                       56            5A  C0 00706         ADDL2   INDENT, SIZE                               ;
                                       57            5A  D0 00709         MOVL    INDENT, LINE_SIZE                          ;
                                       58     FF00   CD  9A 0070C 66$:    MOVZBL  LOCAL_ACE, R8                              ;
                                       57       FC A847  9E 00711         MOVAB   -4(R8)[LINE_SIZE], LINE_SIZE               ;
                                       50       FC   A8  9E 00716         MOVAB   -4(R8), R0                                 ; 1073
                  0A5C CE46     FF04   CD            50  28 0071A         MOVC3   R0, LOCAL_ACE+4, BUFFER[SIZE]              ; 1075
                                       56       FC A846  9E 00723         MOVAB   -4(R8)[SIZE], SIZE                         ; 1076
                                       2B       28   AE  E9 00728         BLBC    40(SP), 68$                                ; 1077
                                       50       01   A7  9E 0072C         MOVAB   1(R7), R0                                  ;
                                       59            50  D1 00730         CMPL    R0, WIDTH                                  ;
                                                     22  1B 00733         BLEQU   68$                                        ;
                                                     5B  D5 00735         TSTL    TERM_LENGTH                                ;
                                                     0F  15 00737         BLEQ    67$                                        ;
                  0A5C CE46       08   BE            5B  28 00739         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]   ;
                                       50       14   BC  3C 00741         MOVZWL  @TERM_DESC, R0                             ;
                                       56            50  C0 00745         ADDL2   R0, SIZE                                   ;
           5A            20            6E            00  2C 00748 67$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]       ;
                                              0A5C CE46     0074D                                                            ;
                                       56            5A  C0 00751         ADDL2   INDENT, SIZE                               ;
                                       57            5A  D0 00754         MOVL    INDENT, LINE_SIZE                          ;
                                                     57  D6 00757 68$:    INCL    LINE_SIZE                                  ;
                                0A5C CE46 00000000'  EF  90 00759         MOVB    P.AD0, BUFFER[SIZE]                        ;
                                                   0120  31 00763         BRW     80$                                        ;
                                  20   AE            01  90 00766 69$:    MOVB    #1, ACCESS_MASK                            ; 1082
                                  24   AE            01  90 0076A         MOVB    #1, AUDIT_MASK                             ; 1083
                                       05       18   AE  91 0076E         CMPB    24(SP), #5                                 ; 1084
                                       40            12  00772            BNEQ    72$                                        ;
                                       2B       28   AE  E9 00774         BLBC    40(SP), 71$                                ; 1085
                                       50       0E   A7  9E 00778         MOVAB   14(R7), R0                                 ;
                                       59            50  D1 0077C         CMPL    R0, WIDTH                                  ;
                                                     22  1B 0077F         BLEQU   71$                                        ;
                                                     5B  D5 00781         TSTL    TERM_LENGTH                                ;
                                                     0F  15 00783         BLEQ    70$                                        ;
                  0A5C CE46       08   BE            5B  28 00785         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]   ;
                                       50       14   BC  3C 0078D         MOVZWL  @TERM_DESC, R0                             ;
                                       56            50  C0 00791         ADDL2   R0, SIZE                                   ;
           5A            20            6E            00  2C 00794 70$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]       ;
                                              0A5C CE46     00799                                                            ;
                                       56            5A  C0 0079D         ADDL2   INDENT, SIZE                               ;
                                       57            5A  D0 007A0         MOVL    INDENT, LINE_SIZE                          ;
                                       57            0E  C0 007A3 71$:    ADDL2   #14, LINE_SIZE                             ;
                  0A5C CE46 00000000'  EF            0E  28 007A6         MOVC3   #14, P.ADP, BUFFER[SIZE]                   ;
                                       56            0E  C0 007B1         ADDL2   #14, SIZE                                  ;
                                       06       18   AE  91 007B4 72$:    CMPB    24(SP), #6                                 ; 1086
                                       40            12  007B8            BNEQ    75$                                        ;
                                       2B       28   AE  E9 007BA         BLBC    40(SP), 74$                                ; 1087
                                       50       0E   A7  9E 007BE         MOVAB   14(R7), R0                                 ;
                                       59            50  D1 007C2         CMPL    R0, WIDTH                                  ;
                                                     22  1B 007C5         BLEQU   74$                                        ;
                                                     5B  D5 007C7         TSTL    TERM_LENGTH                                ;
                                                     0F  15 007C9         BLEQ    73$                                        ;
                  0A5C CE46       08   BE            5B  28 007CB         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]   ;
                                       50       14   BC  3C 007D3         MOVZWL  @TERM_DESC, R0                             ;
                                       56            50  C0 007D7         ADDL2   R0, SIZE                                   ;
           5A            20            6E            00  2C 007DA 73$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]       ;
```

```
                                                             C  1
SYSACLSRV                                          16-Sep-1984 01:51:51    VAX-11 Bliss-32 V4.0-742           Page  58
V04-000        $FORMAT_ACL system service          14-Sep-1984 12:40:53    [LOADSS.SRC]SYSACLSRV.B32;1              (5)

                                 0A5C CE46      007DF                                                      ;
                          56           5A   C0 007E3          ADDL2   INDENT, SIZE                         ;
                          57           5A   D0 007E6          MOVL    INDENT, LINE_SIZE                    ;
                          57           0E   C0 007E9  74$:    ADDL2   #14, LINE_SIZE                       ;
      0A5C CE46 00000000'  EF          0E   28 007EC          MOVC3   #14, P.ADQ, BUFFER[SIZE]             ;
                          56           0E   C0 007F7          ADDL2   #14, SIZE                            ;
                          31     28    AE   E9 007FA  75$:    BLBC    40(SP), 77$                          ;  1088
                          50   FF00    CD   9A 007FE          MOVZBL  LOCAL_ACE, R0                        ;
                          50     F8 A047    9E 00803          MOVAB   -8(R0)[LINE_SIZE], R0                ;
                          59           50   D1 00808          CMPL    R0, WIDTH                            ;
                                       22   1B 0080B          BLEQU   77$                                  ;
                                       5B   D5 0080D          TSTL    TERM_LENGTH                          ;
                                       0F   15 0080F          BLEQ    76$                                  ;
      0A5C CE46       08  BE           5B   28 00811          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] ;
                          50     14    BC   3C 00819          MOVZWL  @TERM_DESC, R0                       ;
                          56           50   C0 0081D          ADDL2   R0, SIZE                             ;
5A              20        6E           00   2C 00820  76$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]  ;
                                 0A5C CE46      00825                                                      ;
                          56           5A   C0 00829          ADDL2   INDENT, SIZE                         ;
                          57           5A   D0 0082C          MOVL    INDENT, LINE_SIZE                    ;
                          58   FF00    CD   9A 0082F  77$:    MOVZBL  LOCAL_ACE, R8                        ;
                          57     F8 A847    9E 00834          MOVAB   -8(R8)[LINE_SIZE], LINE_SIZE         ;
                          50     F8    A8   9E 00839          MOVAB   -8(R8), R0                           ;  1089
      0A5C CE46     FF08  CD           50   28 0083D          MOVC3   R0, LOCAL_ACE+8, BUFFER[SIZE]        ;  1091
                          56     F8 A846    9E 00846          MOVAB   -8(R8)[SIZE], SIZE                   ;  1092
                          2B     28    AE   E9 0084B          BLBC    40(SP), 79$                          ;  1093
                          50     01    A7   9E 0084F          MOVAB   1(R7), R0                            ;
                          59           50   D1 00853          CMPL    R0, WIDTH                            ;
                                       22   1B 00856          BLEQU   79$                                  ;
                                       5B   D5 00858          TSTL    TERM_LENGTH                          ;
                                       0F   15 0085A          BLEQ    78$                                  ;
      0A5C CE46       08  BE           5B   28 0085C          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] ;
                          50     14    BC   3C 00864          MOVZWL  @TERM_DESC, R0                       ;
                          56           50   C0 00868          ADDL2   R0, SIZE                             ;
5A              20        6E           00   2C 0086B  78$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]  ;
                                 0A5C CE46      00870                                                      ;
                          56           5A   C0 00874          ADDL2   INDENT, SIZE                         ;
                          57           5A   D0 00877          MOVL    INDENT, LINE_SIZE                    ;
                                       57   D6 0087A  79$:    INCL    LINE_SIZE                            ;
                 0A5C CE46 00000000'   EF   90 0087C          MOVB    P.ADR, BUFFER[SIZE]                  ;
                                       56   D6 00886  80$:    INCL    SIZE                                 ;
                                     0492   31 00888          BRW     126$                                 ;  1037
                          2B     28    AE   E9 0088B  81$:    BLBC    40(SP), 83$                          ;  1097
                          50     13    A7   9E 0088F          MOVAB   19(R7), R0                           ;
                          59           50   D1 00893          CMPL    R0, WIDTH                            ;
                                       22   1B 00896          BLEQU   83$                                  ;
                                       5B   D5 00898          TSTL    TERM_LENGTH                          ;
                                       0F   15 0089A          BLEQ    82$                                  ;
      0A5C CE46       08  BE           5B   28 0089C          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE] ;
                          50     14    BC   3C 008A4          MOVZWL  @TERM_DESC, R0                       ;
                          56           50   C0 008A8          ADDL2   R0, SIZE                             ;
5A              20        6E           00   2C 008AB  82$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]  ;
                                 0A5C CE46      008B0                                                      ;
                          56           5A   C0 008B4          ADDL2   INDENT, SIZE                         ;
                          57           5A   D0 008B7          MOVL    INDENT, LINE_SIZE                    ;
                          57           13   C0 008BA  83$:    ADDL2   #19, LINE_SIZE                       ;
      0A5C CE46 00000000'  EF          13   28 008BD          MOVC3   #19, P.ADS, BUFFER[SIZE]             ;
```

```
                          56          13  C0 008C8          ADDL2   #19, SIZE                                    ;
                                 1C   AE  D4 008CB          CLRL    K                                            ; 1098
                03        00     1C   AE  CF 008CE 84$:     CASEL   K, #0, #3                                    ; 1101
     0038     0028      0018        0008     008D3 85$:     .WORD   86$-85$,-                                    ;
                                                                    87$-85$,-                                    ;
                                                                    88$-85$,-                                    ;
                                                                    89$-85$                                      ;
                 18   AE      FF08    CD  D0 008DB 86$:     MOVL    LOCAL_ACE+8, PROT_VALUE                      ; 1104
                 04   AE 00000000'    EF  9E 008E1          MOVAB   P.ADT, PROT_FIELD_DSC                        ; 1105
                                      2E  11 008E9          BRB     90$                                          ; 1101
                 18   AE      FF0C    CD  D0 008EB 87$:     MOVL    LOCAL_ACE+12, PROT_VALUE                     ; 1108
                 04   AE 00000000'    EF  9E 008F1          MOVAB   P.ADV, PROT_FIELD_DSC                        ; 1109
                                      1E  11 008F9          BRB     90$                                          ; 1101
                 18   AE      FF10    CD  D0 008FB 88$:     MOVL    LOCAL_ACE+16, PROT_VALUE                     ; 1112
                 04   AE 00000000'    EF  9E 00901          MOVAB   P.ADX, PROT_FIELD_DSC                        ; 1113
                                      0E  11 00909          BRB     90$                                          ; 1101
                 18   AE      FF14    CD  D0 0090B 89$:     MOVL    LOCAL_ACE+20, PROT_VALUE                     ; 1116
                 04   AE 00000000'    EF  9E 00911          MOVAB   P.ADZ, PROT_FIELD_DSC                        ; 1117
                                      58  D4 00919 90$:     CLRL    PROT_IDX                                     ; 1120
                                      50  D4 0091B          CLRL    J                                            ; 1121
                      51 00000000'EF40    9A 0091D 91$:     MOVZBL  PROT_CODE[J], R1                             ; 1124
                                      0D  13 00925          BEQL    92$                                          ;
           08         18      AE      50  E0 00927          BBS     J, PROT_VALUE, 92$                           ;
                    FED8 CD48         51  90 0092C          MOVB    R1, PROT_BUF[PROT_IDX]                       ; 1127
                                      58  D6 00932          INCL    PROT_IDX                                     ; 1128
           E5         50              1F  F3 00934 92$:     AOBLEQ  #31, J, 91$                                  ; 1121
                      31         28   AE  E9 00938          BLBC    40(SP), 94$                                  ; 1131
                      50         04   BE  3C 0093C          MOVZWL  @PROT_FIELD_DSC, R0                          ;
                      50              57  C0 00940          ADDL2   LINE_SIZE, R0                                ;
                      50              58  C0 00943          ADDL2   PROT_IDX, R0                                 ;
                      59              50  D1 00946          CMPL    R0, WIDTH                                    ;
                                      22  1B 00949          BLEQU   94$                                          ;
                                      5B  D5 0094B          TSTL    TERM_LENGTH                                  ;
                                      0F  15 0094D          BLEQ    93$                                          ;
      0A5C CE46       08      BE      5B  28 0094F          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]     ;
                      50         14   BC  3C 00957          MOVZWL  @TERM_DESC, R0                               ;
                      56              50  C0 0095B          ADDL2   R0, SIZE                                     ;
     5A       20      6E              00  2C 0095E 93$:     MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]          ;
                                0A5C CE46    00963                                                       ;
                      56              5A  C0 00967          ADDL2   INDENT, SIZE                                 ;
                      57              5A  D0 0096A          MOVL    INDENT, LINE_SIZE                            ;
                 10   AE         04   BE  3C 0096D 94$:     MOVZWL  @PROT_FIELD_DSC, 16(SP)                      ;
           50         57         10   AE  C1 00972          ADDL3   16(SP), LINE_SIZE, R0                        ;
           57         50              58  C1 00977          ADDL3   PROT_IDX, R0, LINE_SIZE                      ;
        0C   AE 00000200      8F      56  C3 0097B          SUBL3   SIZE, #512, 12(SP)                           ; 1135
                 14   AE    0A5C CE46     9E 00984          MOVAB   BUFFER[SIZE], 20(SP)                         ; 1136
                 7E   04      AE      04  C1 0098B          ADDL3   #4, PROT_FIELD_DSC, -(SP)                    ;
                                      9E  DD 00990          PUSHL   @(SP)+                                       ;
  0C   AE         00          9E      14  AE  2C 00992      MOVC5   20(SP), @(SP)+, #0, 12(SP), @20(SP)          ;
                                      14  BE     00999                                                       ;
                                      14  18 0099B          BGEQ    95$                                          ;
                 14   AE         10   AE  C0 0099D          ADDL2   16(SP), 20(SP)                               ;
                 0C   AE         10   AE  C2 009A2          SUBL2   16(SP), 12(SP)                               ;
  0C   AE         00      FED8    CD  58  2C 009A7          MOVC5   PROT_IDX, PROT_BUF, #0, 12(SP), @20(SP)      ;
                                 14   BE     009AF                                                       ;
                 50   56         10   AE  C1 009B1 95$:     ADDL3   16(SP), SIZE, R0                             ; 1137
                 56   50              58  C1 009B6          ADDL3   PROT_IDX, R0, SIZE                           ;
```

```
                                  2B        28   AE   E9 009BA         BLBC    40(SP), 97$                              ; 1138
                                  50        01   A7   9E 009BE         MOVAB   1(R7), R0
                                  59             50   D1 009C2         CMPL    R0, WIDTH
                                                 22   1B 009C5         BLEQU   97$
                                                 5B   D5 009C7         TSTL    TERM_LENGTH
                                                 0F   15 009C9         BLEQ    96$
              0A5C CE46      08   BE             5B   28 009CB         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]
                                  50        14   BC   3C 009D3         MOVZWL  @TERM_DESC, R0
                                  56             50   C0 009D7         ADDL2   R0, SIZE
         5A          20           6E             00   2C 009DA  96$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]
                                       0A5C CE46        009DF
                                  56             5A   C0 009E3         ADDL2   INDENT, SIZE
                                  57             5A   D0 009E6         MOVL    INDENT, LINE_SIZE
                                                 57   D6 009E9  97$:   INCL    LINE_SIZE
                        0A5C CE46 00000000'  EF  90 009EB              MOVB    P.AEB, BUFFER[SIZE]
                                                 56   D6 009F5         INCL    SIZE
       FEDO       1C   AE         01             03   F1 009F7         ACBL    #3, #1, K, 84$                           ; 1098
                                               031C   31 009FE         BRW     126$                                     ; 1037
                                  2B        28   AE   E9 00A01  98$:   BLBC    40(SP), 100$                             ; 1143
                                  50        0F   A7   9E 00A05         MOVAB   15(R7), R0
                                  59             50   D1 00A09         CMPL    R0, WIDTH
                                                 22   1B 00A0C         BLEQU   100$
                                                 5B   D5 00A0E         TSTL    TERM_LENGTH
                                                 0F   15 00A10         BLEQ    99$
              0A5C CE46      08   BE             5B   28 00A12         MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]
                                  50        14   BC   3C 00A1A         MOVZWL  @TERM_DESC, R0
                                  56             50   C0 00A1E         ADDL2   R0, SIZE
         5A          20           6E             00   2C 00A21  99$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]
                                       0A5C CE46        00A26
                                  56             5A   C0 00A2A         ADDL2   INDENT, SIZE
                                  57             5A   D0 00A2D         MOVL    INDENT, LINE_SIZE
                                  57             0F   C0 00A30  100$:  ADDL2   #15, LINE_SIZE
              0A5C CE46 00000000' EF             0F   28 00A33         MOVC3   #15, P.AEC, BUFFER[SIZE]
                                  56             0F   C0 00A3E         ADDL2   #15, SIZE
         08          00           6E             00   2C 00A41         MOVC5   #0, (SP), #0, #8, VOLNAM_DESC            ; 1144
                                          084C   CE      00A46
         12          00           6E             00   2C 00A49         MOVC5   #0, (SP), #0, #18, VOLNAM_TEXT           ; 1145
                                          0838   CE      00A4E
         08          00           6E             00   2C 00A51         MOVC5   #0, (SP), #0, #8, FILENAME_DESC          ; 1146
                                          0830   CE      00A56
    0200 8F          00           6E             00   2C 00A59         MOVC5   #0, (SP), #0, #512, FILENAME_TEXT        ; 1147
                                            30   AE      00A60
              0838   CE 00000000' EF             05   28 00A62         MOVC3   #5, P.AED, VOLNAM_TEXT                   ; 1149
         0D          00      FF04 CD             0C   2C 00A6C         MOVC5   #12, LOCAL_ACE+4, #0, #13, (R3)
                                                 63      00A73
              0838   CE           12             00   3A 00A74         LOCC    #0, #18, VOLNAM_TEXT                     ; 1151
                                                 02   12 00A7A         BNEQ    101$
                                                 51   D4 00A7C         CLRL    R1
                                  50      0838   CE   9E 00A7E  101$:  MOVAB   VOLNAM_TEXT, R0
              084C   CE           51             50   A3 00A83         SUBW3   R0, R1, VOLNAM_DESC
                             0850 CE      0838   CE   9E 00A89         MOVAB   VOLNAM_TEXT, VOLNAM_DESC+4               ; 1153
                             0830 CE      0200   8F   B0 00A90         MOVW    #512, FILENAME_DESC                      ; 1154
                             0834 CE        30   AE   9E 00A97         MOVAB   FILENAME_TEXT, FILENAME_DESC+4           ; 1155
                                          0830   CE   9F 00A9D         PUSHAB  FILENAME_DESC                            ; 1156
                                          0834   CE   9F 00AA1         PUSHAB  FILENAME_DESC
                                          FF10   CD   9F 00AA5         PUSHAB  LOCAL_ACE+16
                                          0858   CE   9F 00AA9         PUSHAB  VOLNAM_DESC
```

```
SYSACLSRV                                               F  1
V04-000         $FORMAT_ACL system service              16-Sep-1984 01:51:51    VAX-11 Bliss-32 V4.0-742            Page  61
                                                        14-Sep-1984 12:40:53    [LOADSS.SRC]SYSACLSRV.B32;1               (5)

                 00000000G  00            04  FB 00AAD          CALLS   #4, LIB$FID_TO_NAME                        :
                            58            50  D0 00AB4          MOVL    R0, LOCAL_STATUS                           :
                            2B        28  AE  E9 00AB7          BLBC    40(SP), 103$                               : 1158
                            50        0F  A7  9E 00ABB          MOVAB   15(R7), R0                                 :
                            59            50  D1 00ABF          CMPL    R0, WIDTH                                  :
                                          22  1B 00AC2          BLEQU   103$                                       :
                                          5B  D5 00AC4          TSTL    TERM_LENGTH                                :
                                          0F  15 00AC6          BLEQ    102$                                       :
        0A5C CE46        08  BE           5B  28 00AC8          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]   :
                            50        14  BC  3C 00AD0          MOVZWL  @TERM_DESC, R0                             :
                            56            50  C0 00AD4          ADDL2   R0, SIZE                                   :
5A               20         6E            00  2C 00AD7 102$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]        :
                                   0A5C CE46     00ADC                                                             :
                            56            5A  C0 00AE0          ADDL2   INDENT, SIZE                               :
                            57            5A  D0 00AE3          MOVL    INDENT, LINE_SIZE                          :
                            57            0F  C0 00AE6 103$:    ADDL2   #15, LINE_SIZE                             :
        0A5C CE46 00000000' EF            0F  28 00AE9          MOVC3   #15, P.AEE, BUFFER[SIZE]                   :
                            56            0F  C0 00AF4          ADDL2   #15, SIZE                                  :
                            03            58  E8 00AF7          BLBS    LOCAL_STATUS, 104$                         : 1159
                                        00E8  31 00AFA          BRW     117$                                       :
                     18     AE      0834  CE  D0 00AFD 104$:    MOVL    FILENAME_DESC+4, SEGMENT_START             : 1164
                 50         59            57  C3 00B03          SUBL3   LINE_SIZE, WIDTH, R0                       : 1165
50      0830     CE         10            00  ED 00B07          CMPZV   #0, #16, FILENAME_DESC, R0                 :
                                          05  1E 00B0E          BGEQU   105$                                       :
                            50      0830  CE  3C 00B10          MOVZWL  FILENAME_DESC, R0                          :
                            58            50  D0 00B15 105$:    MOVL    R0, SEGMENT_SIZE                           :
                     1C     AE      0830  CE  3C 00B18          MOVZWL  FILENAME_DESC, 28(SP)                      : 1168
                     1C     AE            58  D1 00B1E 106$:    CMPL    SEGMENT_SIZE, 28(SP)                       :
                                          2C  1E 00B22          BGEQU   110$                                       :
                            50        01  A8  9E 00B24          MOVAB   1(R8), J                                   : 1173
                                          23  11 00B28          BRB     109$                                       :
                            52        18  AE  D0 00B2A 107$:    MOVL    SEGMENT_START, R2                          :
                            51     FF A042  9A 00B2E            MOVZBL  -1(J)[R2], R1                              :
                            3A            51  91 00B33          CMPB    R1, #58                                    :
                                          10  13 00B36          BEQL    108$                                       :
                     5D     8F            51  91 00B38          CMPB    R1, #93                                    : 1174
                                          0A  13 00B3C          BEQL    108$                                       :
                            2E            51  91 00B3E          CMPB    R1, #46                                    : 1175
                                          05  13 00B41          BEQL    108$                                       :
                            3B            51  91 00B43          CMPB    R1, #59                                    : 1176
                                          05  12 00B46          BNEQ    109$                                       :
                            58            50  D0 00B48 108$:    MOVL    J, SEGMENT_SIZE                            : 1179
                                          03  11 00B4B          BRB     110$                                       : 1178
                            DA            50  F5 00B4D 109$:    SOBGTR  J, 107$                                    : 1170
        0A5C CE46        18  BE           58  28 00B50 110$:    MOVC3   SEGMENT_SIZE, @SEGMENT_START, BUFFER[SIZE] : 1183
                            57            58  C0 00B58          ADDL2   SEGMENT_SIZE, LINE_SIZE                    : 1184
                            56            58  C0 00B5B          ADDL2   SEGMENT_SIZE, SIZE                         : 1185
                   0830     CE            58  A2 00B5E          SUBW2   SEGMENT_SIZE, FILENAME_DESC                : 1186
                     18     AE            58  C0 00B63          ADDL2   SEGMENT_SIZE, SEGMENT_START                : 1187
                     1C     AE      0830  CE  3C 00B67          MOVZWL  FILENAME_DESC, 28(SP)                      : 1188
                                          1E  15 00B6D          BLEQ    112$                                       :
                                          5B  D5 00B6F          TSTL    TERM_LENGTH                                :
                                          0B  15 00B71          BLEQ    111$                                       :
        0A5C CE46        08  BE           5B  28 00B73          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]   :
                            56            5B  C0 00B7B          ADDL2   TERM_LENGTH, SIZE                          :
5A               20         6E            00  2C 00B7E 111$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]        :
                                   0A5C CE46     00B83                                                             :
```

```
                             56            5A  C0 00B87          ADDL2   INDENT, SIZE                                      ;
                             57            5A  D0 00B8A          MOVL    INDENT, LINE_SIZE                                 ;
              50             59            57  C3 00B8D 112$:    SUBL3   LINE_SIZE, WIDTH, R0                              ; 1189
                     1C      AE            50  D1 00B91          CMPL    R0, 28(SP)                                        ;
                                           04  1B 00B95          BLEQU   113$                                              ;
                             50      1C    AE  D0 00B97          MOVL    28(SP), R0                                        ;
                             58            50  D0 00B9B 113$:    MOVL    R0, SEGMENT_SIZE                                  ;
                                     1C    AE  D5 00B9E          TSTL    28(SP)                                            ; 1191
                                           03  15 00BA1          BLEQ    114$                                              ;
                                         FF78  31 00BA3          BRW     106$                                              ;
                             2B      28    AE  E9 00BA6 114$:    BLBC    40(SP), 116$                                      ; 1192
                             50      01    A7  9E 00BAA          MOVAB   1(R7), R0                                         ;
                             59            50  D1 00BAE          CMPL    R0, WIDTH                                         ;
                                           22  1B 00BB1          BLEQU   116$                                              ;
                                           5B  D5 00BB3          TSTL    TERM_LENGTH                                       ;
                                           0F  15 00BB5          BLEQ    115$                                              ;
          0A5C CE46          08      BE    5B  28 00BB7          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]          ;
                             50      14    BC  3C 00BBF          MOVZWL  @TERM_DESC, R0                                    ;
                             56            50  C0 00BC3          ADDL2   R0, SIZE                                          ;
   5A             20         6E            00  2C 00BC6 115$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]               ;
                                    0A5C CE46     00BCB                                                                    ;
                             56            5A  C0 00BCF          ADDL2   INDENT, SIZE                                      ;
                             57            5A  D0 00BD2          MOVL    INDENT, LINE_SIZE                                 ;
                                           57  D6 00BD5 116$:    INCL    LINE_SIZE                                         ;
                 0A5C CE46 00000000'       EF  90 00BD7          MOVB    P.AEF, BUFFER[SIZE]                               ;
                                           56  D6 00BE1          INCL    SIZE                                              ;
                                           76  11 00BE3          BRB     120$                                              ; 1159
                      0A54   CE    0200    8F  B0 00BE5 117$:    MOVW    #512, FAO_DESC                                    ; 1196
                      0A58   CE    0854    CE  9E 00BEC          MOVAB   FAO_BUF, FAO_DESC+4                               ; 1197
                                   FF14    CD  DD 00BF3          PUSHL   LOCAL_ACE+20                                      ; 1203
                                   FF12    CD  DD 00BF7          PUSHL   LOCAL_ACE+18                                      ;
                                   FF10    CD  DD 00BFB          PUSHL   LOCAL_ACE+16                                      ;
                                   0A60    CE  9F 00BFF          PUSHAB  FAO_DESC                                          ;
                                   0A64    CE  9F 00C03          PUSHAB  FAO_DESC                                          ;
                              00000000'    EF  9F 00C07          PUSHAB  P.AEG                                             ;
                 00000000G   00            06  FB 00C0D          CALLS   #6, SYS$FAO                                       ;
                             2F      28    AE  E9 00C14          BLBC    40(SP), 119$                                      ; 1204
                             50    0A54    CE  3C 00C18          MOVZWL  FAO_DESC, R0                                      ;
                             50            57  C0 00C1D          ADDL2   LINE_SIZE, R0                                     ;
                             59            50  D1 00C20          CMPL    R0, WIDTH                                         ;
                                           22  1B 00C23          BLEQU   119$                                              ;
                                           5B  D5 00C25          TSTL    TERM_LENGTH                                       ;
                                           0F  15 00C27          BLEQ    118$                                              ;
          0A5C CE46          08      BE    5B  28 00C29          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]          ;
                             50      14    BC  3C 00C31          MOVZWL  @TERM_DESC, R0                                    ;
                             56            50  C0 00C35          ADDL2   R0, SIZE                                          ;
   5A             20         6E            00  2C 00C38 118$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]               ;
                                    0A5C CE46     00C3D                                                                    ;
                             56            5A  C0 00C41          ADDL2   INDENT, SIZE                                      ;
                             57            5A  D0 00C44          MOVL    INDENT, LINE_SIZE                                 ;
                             58    0A54    CE  3C 00C47 119$:    MOVZWL  FAO_DESC, R8                                      ;
                             57            58  C0 00C4C          ADDL2   R8, LINE_SIZE                                     ;
          0A5C CE46        0A58      DE    58  28 00C4F          MOVC3   R8, @FAO_DESC+4, BUFFER[SIZE]                     ; 1205
                             56            58  C0 00C58          ADDL2   R8, SIZE                                          ; 1206
                             2B      28    AE  E9 00C5B 120$:    BLBC    40(SP), 122$                                      ; 1208
                             50      16    A7  9E 00C5F          MOVAB   22(R7), R0                                        ;
                             59            50  D1 00C63          CMPL    R0, WIDTH                                         ;
```

```
                                          22   1B 00C66          BLEQU   122$                                        ;
                                          5B   D5 00C68          TSTL    TERM_LENGTH                                 ;
                                          0F   15 00C6A          BLEQ    121$                                        ;
         0A5C CE46      08     BE         5B   28 00C6C          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    ;
                               50    14   BC   3C 00C74          MOVZWL  @TERM_DESC, R0                              ;
                               56         50   C0 00C78          ADDL2   R0, SIZE                                    ;
   5A         20               6E         00   2C 00C7B  121$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         ;
                                    0A5C CE46     00C80                                                              ;
                               56         5A   C0 00C84          ADDL2   INDENT, SIZE                                ;
                               57         5A   D0 00C87          MOVL    INDENT, LINE_SIZE                           ;
                               57         16   C0 00C8A  122$:   ADDL2   #22, LINE_SIZE                              ;
         0A5C CE46 00000000'   EF         16   28 00C8D          MOVC3   #22, P.AEI, BUFFER[SIZE]                    ;
                               56         16   C0 00C98          ADDL2   #22, SIZE                                   ;
                       0A54    CE  0200   8F   B0 00C9B          MOVW    #512, FAO_DESC                              ; 1209
                       0A58    CE  0854   CE   9E 00CA2          MOVAB   FAO_BUF, FAO_DESC+4                         ; 1210
                                   FF18   CD   9F 00CA9          PUSHAB  LOCAL_ACE+24                                ; 1214
                                   0A58   CE   9F 00CAD          PUSHAB  FAO_DESC                                    ;
                                   0A5C   CE   9F 00CB1          PUSHAB  FAO_DESC                                    ;
                               00000000'  EF   9F 00CB5          PUSHAB  P.AEJ                                       ;
                  00000000G    00         04   FB 00CBB          CALLS   #4, SYS$FAO                                 ;
                       085F    CE         3A   90 00CC2          MOVB    #58, FAO_BUF+11                             ; 1215
                               20  0854   CE   91 00CC7          CMPB    FAO_BUF, #32                                ; 1216
                                          08   12 00CCC          BNEQ    123$                                        ;
                                   0A54   CE   B7 00CCE          DECW    FAO_DESC                                    ; 1219
                                   0A58   CE   D6 00CD2          INCL    FAO_DESC+4                                  ; 1220
                               2F    28   AE   E9 00CD6  123$:   BLBC    40(SP), 125$                                ; 1222
                               50  0A54   CE   3C 00CDA          MOVZWL  FAO_DESC, R0                                ;
                               50         57   C0 00CDF          ADDL2   LINE_SIZE, R0                               ;
                               59         50   D1 00CE2          CMPL    R0, WIDTH                                   ;
                                          22   1B 00CE5          BLEQU   125$                                        ;
                                          5B   D5 00CE7          TSTL    TERM_LENGTH                                 ;
                                          0F   15 00CE9          BLEQ    124$                                        ;
         0A5C CE46      08     BE         5B   28 00CEB          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    ;
                               50    14   BC   3C 00CF3          MOVZWL  @TERM_DESC, R0                              ;
                               56         50   C0 00CF7          ADDL2   R0, SIZE                                    ;
   5A         20               6E         00   2C 00CFA  124$:   MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]         ;
                                    0A5C CE46     00CFF                                                              ;
                               56         5A   C0 00D03          ADDL2   INDENT, SIZE                                ;
                               57         5A   D0 00D06          MOVL    INDENT, LINE_SIZE                           ;
                               58  0A54   CE   3C 00D09  125$:   MOVZWL  FAO_DESC, R8                                ;
                               57         58   C0 00D0E          ADDL2   R8, LINE_SIZE                               ;
         0A5C CE46    0A58     DE         58   28 00D11          MOVC3   R8, @FAO_DESC+4, BUFFER[SIZE]               ; 1223
                               56         58   C0 00D1A          ADDL2   R8, SIZE                                    ; 1224
                               58  FF02   CD   B0 00D1D  126$:   MOVW    LOCAL_ACE+2, FLAGS                          ; 1293
                               03    24   AE   E9 00D22          BLBC    AUDIT_MASK, 127$                            ; 1294
                               58         03   8A 00D26          BICB2   #3, FLAGS                                   ; 1295
                                          58   B5 00D29  127$:   TSTW    FLAGS                                       ; 1296
                                          03   12 00D2B          BNEQ    128$                                        ;
                                   01B6   31 00D2D               BRW     146$                                        ;
                               2B    28   AE   E9 00D30  128$:   BLBC    40(SP), 130$                                ; 1299
                               50    08   A7   9E 00D34          MOVAB   8(R7), R0                                   ;
                               59         50   D1 00D38          CMPL    R0, WIDTH                                   ;
                                          22   1B 00D3B          BLEQU   130$                                        ;
                                          5B   D5 00D3D          TSTL    TERM_LENGTH                                 ;
                                          0F   15 00D3F          BLEQ    129$                                        ;
         0A5C CE46      08     BE         5B   28 00D41          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]    ;
                               50    14   BC   3C 00D49          MOVZWL  @TERM_DESC, R0                              ;
```

```
                              56        50  C0 00D4D          ADDL2   RO, SIZE                                          ;
      5A        20            6E        00  2C 00D50 129$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]               ;
                                  0A5C CE46    00D55                                                                    ;
                              56        5A  C0 00D59          ADDL2   INDENT, SIZE                                      ;
                              57        5A  D0 00D5C          MOVL    INDENT, LINE_SIZE                                 ;
                              57        08  C0 00D5F 130$:    ADDL2   #8, LINE_SIZE                                     ;
           0A5C CE46 00000000'  EF      08  28 00D62          MOVC3   #8, P.AFA, BUFFER[SIZE]                           ;
                              56        08  C0 00D6D          ADDL2   #8, SIZE                                          ;
                40            58        08  E5 00D70          BBCC    #8, FLAGS, 133$                                   ; 1300
                              2B    28  AE  E9 00D74          BLBC    40(SP), 132$                                      ; 1301
                              50    08  A7  9E 00D78          MOVAB   8(R7), RO                                         ;
                              59        50  D1 00D7C          CMPL    RO, WIDTH                                         ;
                              22        1B  00D7F             BLEQU   132$                                              ;
                              5B        D5  00D81             TSTL    TERM_LENGTH                                       ;
                              0F        15  00D83             BLEQ    131$                                              ;
           0A5C CE46        08  BE      5B  28 00D85          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]          ;
                              50    14  BC  3C 00D8D          MOVZWL  @TERM_DESC, RO                                    ;
                              56        50  C0 00D91          ADDL2   RO, SIZE                                          ;
      5A        20            6E        00  2C 00D94 131$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]               ;
                                  0A5C CE46    00D99                                                                    ;
                              56        5A  C0 00D9D          ADDL2   INDENT, SIZE                                      ;
                              57        5A  D0 00DA0          MOVL    INDENT, LINE_SIZE                                 ;
                              57        08  C0 00DA3 132$:    ADDL2   #8, LINE_SIZE                                     ;
           0A5C CE46 00000000'  EF      08  28 00DA6          MOVC3   #8, P.AFB, BUFFER[SIZE]                           ;
                              56        08  C0 00DB1          ADDL2   #8, SIZE                                          ;
                40            58        0A  E5 00DB4 133$:    BBCC    #10, FLAGS, 136$                                  ; 1302
                              2B    28  AE  E9 00DB8          BLBC    40(SP), 135$                                      ; 1303
                              50    07  A7  9E 00DBC          MOVAB   7(R7), RO                                         ;
                              59        50  D1 00DC0          CMPL    RO, WIDTH                                         ;
                              22        1B  00DC3             BLEQU   135$                                              ;
                              5B        D5  00DC5             TSTL    TERM_LENGTH                                       ;
                              0F        15  00DC7             BLEQ    134$                                              ;
           0A5C CE46        08  BE      5B  28 00DC9          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]          ;
                              50    14  BC  3C 00DD1          MOVZWL  @TERM_DESC, RO                                    ;
                              56        50  C0 00DD5          ADDL2   RO, SIZE                                          ;
      5A        20            6E        00  2C 00DD8 134$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]               ;
                                  0A5C CE46    00DDD                                                                    ;
                              56        5A  C0 00DE1          ADDL2   INDENT, SIZE                                      ;
                              57        5A  D0 00DE4          MOVL    INDENT, LINE_SIZE                                 ;
                              57        07  C0 00DE7 135$:    ADDL2   #7, LINE_SIZE                                     ;
           0A5C CE46 00000000'  EF      07  28 00DEA          MOVC3   #7, P.AFC, BUFFER[SIZE]                           ;
                              56        07  C0 00DF5          ADDL2   #7, SIZE                                          ;
                40            58        09  E5 00DF8 136$:    BBCC    #9, FLAGS, 139$                                   ; 1304
                              2B    28  AE  E9 00DFC          BLBC    40(SP), 138$                                      ; 1305
                              50    0A  A7  9E 00E00          MOVAB   10(R7), RO                                        ;
                              59        50  D1 00E04          CMPL    RO, WIDTH                                         ;
                              22        1B  00E07             BLEQU   138$                                              ;
                              5B        D5  00E09             TSTL    TERM_LENGTH                                       ;
                              0F        15  00E0B             BLEQ    137$                                              ;
           0A5C CE46        08  BE      5B  28 00E0D          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]          ;
                              50    14  BC  3C 00E15          MOVZWL  @TERM_DESC, RO                                    ;
                              56        50  C0 00E19          ADDL2   RO, SIZE                                          ;
      5A        20            6E        00  2C 00E1C 137$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]               ;
                                  0A5C CE46    00E21                                                                    ;
                              56        5A  C0 00E25          ADDL2   INDENT, SIZE                                      ;
                              57        5A  D0 00E28          MOVL    INDENT, LINE_SIZE                                 ;
                              57        0A  C0 00E2B 138$:    ADDL2   #10, LINE_SIZE                                    ;
```

```
        0A5C CE46 00000000'  EF           0A  28 00E2E        MOVC3   #10, P.AFD, BUFFER[SIZE]                       ;
                             56           0A  C0 00E39        ADDL2   #10, SIZE                                      ;
                  40         58           0B  E5 00E3C 139$:  BBCC    #11, FLAGS, 142$                               ; 1306
                             2B      28   AE  E9 00E40        BLBC    40(SP), 141$                                   ; 1307
                             50      0C   A7  9E 00E44        MOVAB   12(R7), R0                                     ;
                             59           50  D1 00E48        CMPL    R0, WIDTH                                      ;
                                          22  1B 00E4B        BLEQU   141$                                           ;
                                          5B  D5 00E4D        TSTL    TERM_LENGTH                                    ;
                                          0F  15 00E4F        BLEQ    140$                                           ;
        0A5C CE46       08   BE           5B  28 00E51        MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]       ;
                             50      14   BC  3C 00E59        MOVZWL  @TERM_DESC, R0                                 ;
                             56           50  C0 00E5D        ADDL2   R0, SIZE                                       ;
   5A             20         6E           00  2C 00E60 140$:  MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]            ;
                                   0A5C CE46     00E65                                                               ;
                             56           5A  C0 00E69        ADDL2   INDENT, SIZE                                   ;
                             57           5A  D0 00E6C        MOVL    INDENT, LINE_SIZE                              ;
                             57           0C  C0 00E6F 141$:  ADDL2   #12, LINE_SIZE                                 ;
        0A5C CE46 00000000'  EF           0C  28 00E72        MOVC3   #12, P.AFE, BUFFER[SIZE]                       ;
                             56           0C  C0 00E7D        ADDL2   #12, SIZE                                      ;
                                          58  B5 00E80 142$:  TSTW    FLAGS                                          ; 1308
                                          5C  13 00E82        BEQL    145$                                           ;
                             2B      28   AE  E9 00E84        BLBC    40(SP), 144$                                   ; 1311
                             50      07   A7  9E 00E88        MOVAB   7(R7), R0                                      ;
                             59           50  D1 00E8C        CMPL    R0, WIDTH                                      ;
                                          22  1B 00E8F        BLEQU   144$                                           ;
                                          5B  D5 00E91        TSTL    TERM_LENGTH                                    ;
                                          0F  15 00E93        BLEQ    143$                                           ;
        0A5C CE46       08   BE           5B  28 00E95        MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]       ;
                             50      14   BC  3C 00E9D        MOVZWL  @TERM_DESC, R0                                 ;
                             56           50  C0 00EA1        ADDL2   R0, SIZE                                       ;
   5A             20         6E           00  2C 00EA4 143$:  MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]            ;
                                   0A5C CE46     00EA9                                                               ;
                             56           5A  C0 00EAD        ADDL2   INDENT, SIZE                                   ;
                             57           5A  D0 00EB0        MOVL    INDENT, LINE_SIZE                              ;
                             57           07  C0 00EB3 144$:  ADDL2   #7, LINE_SIZE                                  ;
                       FEF8  CD           07  B0 00EB6        MOVW    #7, FAO_DESCR                                  ; 1312
                       FEFC  CD  0A5C CE46    9E 00EBB        MOVAB   BUFFER[SIZE], FAO_DESCR+4                      ; 1313
                             7E           58  3C 00EC3        MOVZWL  FLAGS, -(SP)                                   ; 1317
                                   FEF8   CD  9F 00EC6        PUSHAB  FAO_DESCR                                      ;
                                   FEF8   CD  9F 00ECA        PUSHAB  FAO_DESCR                                      ;
                              00000000'   EF  9F 00ECE        PUSHAB  P.AFF                                          ;
                  00000000G  00           04  FB 00ED4        CALLS   #4, SYS$FAO                                    ;
                             56           07  C0 00EDB        ADDL2   #7, SIZE                                       ; 1318
                                          06  11 00EDE        BRB     146$                                           ; 1308
                  0A5B CE46               2C  90 00EE0 145$:  MOVB    #44, BUFFER-1[SIZE]                            ; 1320
                             03      20   AE  E8 00EE6 146$:  BLBS    ACCESS_MASK, 147$                              ; 1325
                                   01D6       31 00EEA        BRW     171$                                           ;
                       FF04  CD               D5 00EED 147$:  TSTL    LOCAL_ACE+4                                    ; 1328
                             12               12 00EF1        BNEQ    150$                                           ;
                  03         24   AE          E8 00EF3        BLBS    AUDIT_MASK, 149$                               ; 1329
                                   0189       31 00EF7 148$:  BRW     168$                                           ;
                  06       FF02   CD          E8 00EFA 149$:  BLBS    LOCAL_ACE+2, 150$                              ; 1330
           F2       FF02     CD           01  E1 00EFF        BBC     #1, LOCAL_ACE+2, 148$                          ; 1331
                             2B      28   AE  E9 00F05 150$:  BLBC    40(SP), 152$                                   ; 1334
                             50      07   A7  9E 00F09        MOVAB   7(R7), R0                                      ;
                             59           50  D1 00F0D        CMPL    R0, WIDTH                                      ;
                                          22  1B 00F10        BLEQU   152$                                           ;
                                   5B  D5 00F12          TSTL    TERM_LENGTH                                   :
                                   0F  15 00F14          BLEQ    151$                                          :
        0A5C CE46      08   BE     5B  28 00F16          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]      :
                            50   14   BC  3C 00F1E       MOVZWL  @TERM_DESC, R0                                :
                            56     50  C0 00F22          ADDL2   R0, SIZE                                      :
   5A             20        6E     00  2C 00F25 151$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]           :
                               0A5C CE46  00F2A                                                                :
                            56     5A  C0 00F2E          ADDL2   INDENT, SIZE                                  :
                            57     5A  D0 00F31          MOVL    INDENT, LINE_SIZE                             :
                            57     07  C0 00F34 152$:    ADDL2   #7, LINE_SIZE                                 :
        0A5C CE46 00000000' EF     07  28 00F37          MOVC3   #7, P.AFA, BUFFER[SIZE]                       :
                            56     07  C0 00F42          ADDL2   #7, SIZE                                      :
                                   58  D4 00F45          CLRL    J                                             : 1335
             03      FF04   CD     58  E0 00F47 153$:    BBS     J, LOCAL_ACE+4, 154$                          : 1338
                                 0099  31 00F4D          BRW     160$                                          :
                 50 00000000'   EF     D0 00F50 154$:    MOVL    BIT_NAME_TABLE, R0                            : 1341
                                   28  13 00F57          BEQL    156$                                          :
                                 6048  7F 00F59          PUSHAQ  (R0)[J]                                       : 1344
             9E           08       00  0C 00F5C          PROBER  #0, #8, @(SP)+                                :
                                   1C  13 00F60          BEQL    155$                                          :
                       1C   AE   6048  7E 00F62          MOVAQ   (R0)[J], BIT_NAME_DESC                        : 1345
             54        1C   AE     04  C1 00F67          ADDL3   #4, BIT_NAME_DESC, R4                         : 1347
                            50     64  D0 00F6C          MOVL    (R4), R0                                      :
                            51   1C   BE  3C 00F6F       MOVZWL  @BIT_NAME_DESC, R1                            :
                                   53  D4 00F73          CLRL    R3                                            :
                       00000000G   00  16 00F75          JSB     EXE$PROBER                                    :
                       0C          50  E8 00F7B          BLBS    R0, 157$                                      :
                                 0183  31 00F7E 155$:    BRW     172$                                          : 1349
                       1C   AE 00000000'EF48  D0 00F81 156$:  MOVL  DEFAULT_BITS[J], BIT_NAME_DESC           : 1351
                            30   28   AE  E9 00F8A 157$: BLBC    40(SP), 159$                                  : 1352
                            50   1C   BE  3C 00F8E       MOVZWL  @BIT_NAME_DESC, R0                            :
                            50   01 A047  9E 00F92       MOVAB   1(R0)[LINE_SIZE], R0                          :
                            59     50  D1 00F97          CMPL    R0, WIDTH                                     :
                                   22  1B 00F9A          BLEQU   159$                                          :
                                   5B  D5 00F9C          TSTL    TERM_LENGTH                                   :
                                   0F  15 00F9E          BLEQ    158$                                          :
        0A5C CE46      08   BE     5B  28 00FA0          MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]      :
                            50   14   BC  3C 00FA8       MOVZWL  @TERM_DESC, R0                                :
                            56     50  C0 00FAC          ADDL2   R0, SIZE                                      :
   5A             20        6E     00  2C 00FAF 158$:    MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]           :
                               0A5C CE46  00FB4                                                                :
                            56     5A  C0 00FB8          ADDL2   INDENT, SIZE                                  :
                            57     5A  D0 00FBB          MOVL    INDENT, LINE_SIZE                             :
                       20   AE   1C   BE  3C 00FBE 159$: MOVZWL  @BIT_NAME_DESC, 32(SP)                        :
             50        20   AE     01  C1 00FC3          ADDL3   #1, 32(SP), R0                                :
                            57     50  C0 00FC8          ADDL2   R0, LINE_SIZE                                 :
             7E        1C   AE     04  C1 00FCB          ADDL3   #4, BIT_NAME_DESC, -(SP)                      : 1354
                                   9E  DD 00FD0          PUSHL   @(SP)+                                        :
        0A5C CE46           9E   24   AE  28 00FD2       MOVC3   36(SP), @(SP)+, BUFFER[SIZE]                  :
             50             56   20   AE  C1 00FDA       ADDL3   32(SP), SIZE, R0                              : 1355
                      0A5C CE40    2B  90 00FDF          MOVB    #43, BUFFER[R0]                               :
                            56   01   A0  9E 00FE5       MOVAB   1(R0), SIZE                                   : 1356
 FF58             58        01     1F  F1 00FE9 160$:    ACBL    #31, #1, J, 153$                              : 1335
                            03   24   AE  E8 00FEF       BLBS    AUDIT_MASK, 162$                              : 1359
                                 00CD  31 00FF3 161$:    BRW     171$                                          :
                            40 FF02   CD  E9 00FF6 162$: BLBC    LOCAL_ACE+2, 165$                             : 1362
                            2B   28   AE  E9 00FFB       BLBC    40(SP), 164$                                  :
```

```
                     50       08    A7    9E 00FFF              MOVAB   8(R7), R0                                        ;
                     59             50    D1 01003              CMPL    R0, WIDTH                                        ;
                                    22    1B 01006              BLEQU   164$                                             ;
                                    5B    D5 01008              TSTL    TERM_LENGTH                                      ;
                                    0F    15 0100A              BLEQ    163$                                             ;
        0A5C CE46       08    BE    5B    28 0100C              MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]         ;
                        50    14    BC    3C 01014              MOVZWL  @TERM_DESC, R0                                   ;
                        56          50    C0 01018              ADDL2   R0, SIZE                                         ;
5A              20      6E          00    2C 0101B  163$:       MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]              ;
                            0A5C CE46        01020                                                                       ;
                        56          5A    C0 01024              ADDL2   INDENT, SIZE                                     ;
                        57          5A    D0 01027              MOVL    INDENT, LINE_SIZE                                ;
                        57          08    C0 0102A  164$:       ADDL2   #8, LINE_SIZE                                    ;
        0A5C CE46 00000000'   EF    08    28 0102D              MOVC3   #8, P.AFI, BUFFER[SIZE]                          ;
                        56          08    C0 01038              ADDL2   #8, SIZE                                         ;
              B2      FF02    CD    01    E1 0103B  165$:       BBC     #1, LOCAL_ACE+2, 161$                            ; 1363
                        2B    28    AE    E9 01041              BLBC    40(SP), 167$                                     ;
                        50    08    A7    9E 01045              MOVAB   8(R7), R0                                        ;
                        59          50    D1 01049              CMPL    R0, WIDTH                                        ;
                                    22    1B 0104C              BLEQU   167$                                             ;
                                    5B    D5 0104E              TSTL    TERM_LENGTH                                      ;
                                    0F    15 01050              BLEQ    166$                                             ;
        0A5C CE46       08    BE    5B    28 01052              MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]         ;
                        50    14    BC    3C 0105A              MOVZWL  @TERM_DESC, R0                                   ;
                        56          50    C0 0105E              ADDL2   R0, SIZE                                         ;
5A              20      6E          00    2C 01061  166$:       MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]              ;
                            0A5C CE46        01066                                                                       ;
                        56          5A    C0 0106A              ADDL2   INDENT, SIZE                                     ;
                        57          5A    D0 0106D              MOVL    INDENT, LINE_SIZE                                ;
                        57          08    C0 01070  167$:       ADDL2   #8, LINE_SIZE                                    ;
        0A5C CE46 00000000'   EF    08    28 01073              MOVC3   #8, P.AFJ, BUFFER[SIZE]                          ;
                        56          08    C0 0107E              ADDL2   #8, SIZE                                         ;
                                    40    11 01081              BRB     171$                                             ; 1328
                        2B    28    AE    E9 01083  168$:       BLBC    40(SP), 170$                                     ; 1366
                        50    0C    A7    9E 01087              MOVAB   12(R7), R0                                       ;
                        59          50    D1 0108B              CMPL    R0, WIDTH                                        ;
                                    22    1B 0108E              BLEQU   170$                                             ;
                                    5B    D5 01090              TSTL    TERM_LENGTH                                      ;
                                    0F    15 01092              BLEQ    169$                                             ;
        0A5C CE46       08    BE    5B    28 01094              MOVC3   TERM_LENGTH, @TERM_POINTER, BUFFER[SIZE]         ;
                        50    14    BC    3C 0109C              MOVZWL  @TERM_DESC, R0                                   ;
                        56          50    C0 010A0              ADDL2   R0, SIZE                                         ;
5A              20      6E          00    2C 010A3  169$:       MOVC5   #0, (SP), #32, INDENT, BUFFER[SIZE]              ;
                            0A5C CE46        010A8                                                                       ;
                        56          5A    C0 010AC              ADDL2   INDENT, SIZE                                     ;
                        57          5A    D0 010AF              MOVL    INDENT, LINE_SIZE                                ;
                        57          0C    C0 010B2  170$:       ADDL2   #12, LINE_SIZE                                   ;
        0A5C CE46 00000000'   EF    0C    28 010B5              MOVC3   #12, P.AFK, BUFFER[SIZE]                         ;
                        56          0C    C0 010C0              ADDL2   #12, SIZE                                        ;
                   0A5B CE46        29    90 010C3  171$:       MOVB    #41, BUFFER-1[SIZE]                              ; 1371
           0C     BC            08  00    0C 010C9              PROBER  #0, #8, @ACL_STRING                              ; 1375
                                    34    13 010CE              BEQL    172$                                             ;
              54      0C        AC  04    C1 010D0              ADDL3   #4, ACL_STRING, R4                               ; 1378
                        50          64    D0 010D5              MOVL    (R4), R0                                         ;
                        51          6E    D0 010D8              MOVL    ACL_STRING_LEN, R1                               ;
                                    53    D4 010DB              CLRL    R3                                               ;
                       00000000G    00    16 010DD              JSB     EXE$PROBEW                                       ;
```

```
                                 1E              50  E9 010E3          BLBC    R0, 172$                                      ;
                 57        0C    AC              04  C1 010E6          ADDL3   #4, ACL_STRING, R7                            ; 1379
      6E         00      0A5C    CE              56  2C 010EB          MOVC5   SIZE, BUFFER, #0, ACL_STRING_LEN, @(R7)+      ;
                                                 97     010F2                                                                ;
                                 50        08    AC  D0 010F3          MOVL    ACL_LENGTH, R0                                ; 1383
                                                 0F  13 010F7          BEQL    173$                                          ;
                 60              04              00  0D 010F9          PROBEW  #0, #4, (R0)                                  ; 1384
                                                 05  13 010FD          BEQL    172$                                          ;
                                 60              56  D0 010FF          MOVL    SIZE, (R0)                                    ; 1385
                                                 04  11 01102          BRB     173$                                          ;
                                 50              0C  D0 01104 172$:    MOVL    #12, R0                                       ;
                                                 04     01107          RET                                                   ;
                                 6E              56  D1 01108 173$:    CMPL    SIZE, ACL_STRING_LEN                          ; 1386
                                                 06  15 0110B          BLEQ    175$                                          ;
                                 50      0601    8F  3C 0110D 174$:    MOVZWL  #1537, R0                                     ; 1387
                                                 04     01112          RET                                                   ;
                                 50              01  D0 01113 175$:    MOVL    #1, R0                                        ;
                                                 04     01116          RET                                                   ; 1389

; Routine Size:  4375 bytes,    Routine Base:  $CODE$ + 0201
```

```
1 %SBTTL  '$CHANGE_ACL system service'
1 GLOBAL ROUTINE SYS$CHANGE_ACL (CHANNEL, OBJECT_TYPE, OBJECT_NAME,
1                               ITEM_LIST, ACCESS_MODE, RESERVED, CONTEXT) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine changes (or reads) the ACL associated with any of the
1 !       defined objects within the system.
1 !
1 !       Note:   Since the length of the ACE is part of the data returned to
1 !               the caller,  the return length parameter of the item list
1 !               is unused.
1 !
1 !       There are basically two types of objects that can have their ACL
1 !       twiddled. There are those objects that require an agent to do the work
1 !       (e.g, files by the XQP) and the others where it is simply necessary to
1 !       locate the ACL queue segment list head.  In the latter case, the work
1 !       is done here.
1 !
1 ! CALLING SEQUENCE:
1 !       SYS$CHANGE_ACL (ARG1, ARG2, ARG3, ARG4, ARG5, ARG6, ARG7)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: number of the channel assigned to the object or 0 if
1 !             the object is specified by ARG2 and ARG3
1 !       ARG2: address of an object type code
1 !       ARG3: address of an object name descriptor
1 !       ARG4: address of a list of item descriptors
1 !       ARG5: address of an access mode longword (used to validate the
1 !             item list and I/O status block)
1 !       ARG6: reserved for future use
1 !       ARG7: address of a context longword
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       ARG7:   address of a context longword
1 !
1 ! IMPLICIT OUTPUTS:
1 !       none
1 !
1 ! ROUTINE VALUE:
1 !       SS$_NORMAL - if the requested action completed successfully
1 !       SS$_NOPRIV - if the requestor did not have privileges for the
1 !                    requested action
1 !
1 ! SIDE EFFECTS:
1 !       The context longword is modified as necessary based upon the
1 !       action requested.
1 !
1 !--
1
2 BEGIN
2
```

```
2 MAP
2         CHANNEL          : WORD,
2         OBJECT_NAME      : REF $BBLOCK,
2         ITEM_LIST        : REF BLOCKVECTOR [, ITM$S_ITEM, BYTE];
2
2 LOCAL
2         STATUS,                                        ! Local routine return status
2         STATUS2,                                       ! Temp status, may overwrite STATUS
2         PSL              : $BBLOCK [4],                ! Local copy of PSL
2         LOCAL_OBJTYP,                                  ! Local copy of object type code
2         LOCAL_IOSB       : VECTOR [4, WORD],           ! Local copy of the I/O status block
2         LOCAL_LOCKID,                                  ! Local copy of the lock-id
2         OBJECT_DESC      : VECTOR [2],                 ! Descriptor of object name
2         SHARE            : BYTE,                       ! Whether to allow sharing or not
2         ITEM_COUNT,                                    ! Number of items in the list
2         ITEM_CODE,                                     ! Code from item list entry
2         ITEM_SIZE,                                     ! Size from item list entry
2         ITEM_ADDR,                                     ! Buffer addr from item list entry
2         LOCAL_CHANNEL    : WORD,                       ! Local copy of user's channel
2         IO_CHANNEL       : WORD,                       ! Object's channel
2         FILE_FAB         : $FAB_DECL,                  ! Object file's FAB
2         FILE_NAM         : $NAM_DECL,                  ! Object file's NAMe block
2         FILE_EXP_NAME    : $BBLOCK [NAM$C_MAXRSS],             ! Expanded name storage
2         FILE_RES_NAME    : $BBLOCK [NAM$C_MAXRSS],             ! Resultant name storage
2         FILE_FIB_DESC    : $BBLOCK [DSC$C_S_BLN],              ! File FIB descriptor
2         FILE_FIB         : $BBLOCK [FIB$C_LENGTH],             ! File FIB storage
2         DVI_ATR_LIST     : BLOCKVECTOR [2, ITM$S_ITEM, BYTE],     ! $GETDVI item list
2         ACP_ATR_PTR,                                   ! Pointer into ACP attribute list
2         ACP_ATR_LIST     : REF BLOCKVECTOR [, 8, BYTE],  ! ACP attribute list
2         ACL_TO_ATR_TAB   : VECTOR [MAX_ACL_ATR + 1]      ! ATR$C to ACL$C xlate
2                            INITIAL (0,
2                                     ATR$C_ADDACLENT,
2                                     ATR$C_DELACLENT,
2                                     ATR$C_MODACLENT,
2                                     ATR$C_FNDACLENT,
2                                     ATR$C_FNDACETYP,
2                                     ATR$C_DELETEACL,
2                                     ATR$C_READACL,
2                                     ATR$C_ACLLENGTH,
2                                     ATR$C_READACE),
2         FUNCTION_CODE,                                 ! QIOW function code
2                                                        ! Also, ACL dispatch code
2         CMK_ARG_LIST     : VECTOR [5];                 ! $CMKRNL arg list
2
2 ! See if an access mode parameter was given.
2
2 CHANGE_ACMODE = 0;
2 IF .ACCESS_MODE NEQA 0
2 THEN IF PROBER (%REF (0), %REF (4), .ACCESS_MODE)
2     THEN CHANGE_ACMODE = ..ACCESS_MODE
2     ELSE RETURN SS$_ACCVIO;
2
2 MOVPSL (PSL);
2 CALL_ACMODE = .PSL[PSL$V_PRVMOD];
2 CHANGE_ACMODE = MAXU (.CHANGE_ACMODE, .CALL_ACMODE);
2
2 ! Determine the validity of the access mode parameter.
```

```
2
2 IF .CHANGE_ACMODE GTRU PSL$C_USER THEN RETURN SS$_BADPARAM;
2
2 ! Get the supplied channel, if any, and verify it.
2
2 IO_CHANNEL = LOCAL_CHANNEL = .CHANNEL;
2 IF .IO_CHANNEL NEQ 0
2 THEN
3     BEGIN
3     STATUS = IOC$VERIFYCHAN (.IO_CHANNEL);
3     IF NOT .STATUS THEN RETURN .STATUS;
2     END;
2
2 ! Get the object type code.
2
2 IF .OBJECT_TYPE NEQA 0
3 THEN (IF PROBER (%REF (0), %REF (4), .OBJECT_TYPE)
3       THEN LOCAL_OBJTYP = ..OBJECT_TYPE
3       ELSE RETURN SS$_ACCVIO)
2 ELSE RETURN SS$_INSFARG;
2
2 ! Check the validity of the object type code.
2
2 IF .LOCAL_OBJTYP LSSU MIN_OBJECT_TYPE
2 OR .LOCAL_OBJTYP GTRU MAX_OBJECT_TYPE
2 THEN RETURN SS$_BADPARAM;
2
2 ! Probe the object name if supplied.
2
2 IF .OBJECT_NAME NEQA 0
2 THEN
3     BEGIN
3     IF NOT PROBER (%REF (0), %REF (DSC$C_S_BLN), .OBJECT_NAME)
3     THEN RETURN SS$_ACCVIO;
3     OBJECT_DESC[0] = .OBJECT_NAME[DSC$W_LENGTH];
3     OBJECT_DESC[1] = .OBJECT_NAME[DSC$A_POINTER];
3     IF NOT EXE$PROBER (0, .OBJECT_DESC[0], .OBJECT_DESC[1])
3     THEN RETURN SS$_ACCVIO;
3     END
2 ELSE
3     BEGIN
3     OBJECT_DESC[0] = 0;
3     OBJECT_DESC[1] = 0;
2     END;
2
2 ! Get any value supplied for the context parameter.
2
2 ACL_CONTEXT = 0;
2 IF .CONTEXT NEQA 0
2 THEN IF PROBEW (%REF (0), %REF (4), .CONTEXT)
2       THEN ACL_CONTEXT = ..CONTEXT
2       ELSE RETURN SS$_ACCVIO;
2
2 ! Count the number of items in the item list.
2
2 SHARE = 1;                                             ! Assume shared access
2 INCR J FROM 0
```

```
2 DO IF PROBER (%REF (0), %REF (ITM$S_ITEM), ITEM_LIST[.J, 0,0,0,0])
3     THEN (IF .ITEM_LIST[.J, ITM$W_BUFSIZ] EQL 0
3            THEN
4                BEGIN
4                ITEM_COUNT = .J;
4                EXITLOOP;
4                END
3            ELSE
4                BEGIN
4                IF .ITEM_LIST[.J, ITM$W_ITMCOD] EQL ACL$C_ADDACLENT
4                OR .ITEM_LIST[.J, ITM$W_ITMCOD] EQL ACL$C_DELACLENT
4                OR .ITEM_LIST[.J, ITM$W_ITMCOD] EQL ACL$C_MODACLENT
4                OR .ITEM_LIST[.J, ITM$W_ITMCOD] EQL ACL$C_DELETEACL
4                THEN SHARE = 0;
3                END)
2     ELSE RETURN SS$_ACCVIO;
2
2 ! Initialize all common (to both types of objects) storage.
2
2 CH$FILL (0, 2*ITM$S_ITEM, DVI_ATR_LIST);
2 CH$FILL (0, DSC$C_S_BLN, LOCK_RESNAM);
2 LOCAL_LOCKID = 0;
2
2 ! Set up the lock resource name prefix.
2
2 LOCK_RESNAM[DSC$W_LENGTH] = RSN_S_PREFIX;
2 LOCK_RESNAM[DSC$A_POINTER] = RESNAM_TEXT;
2 CH$COPY (.$BBLOCK [.LOCK_PREFIX[.LOCAL_OBJTYP], DSC$W_LENGTH],
2           .$BBLOCK [.LOCK_PREFIX[.LOCAL_OBJTYP], DSC$A_POINTER],
2           0,
2           RSN_S_PREFIX, RESNAM_TEXT);
2
2 ! If the call is from user mode, take out a lock to form the parent lock ID
2 ! for all ACL locks. This facilitates releasing them at image rundown.
2
2 IF .CALL_ACMODE EQL PSL$C_USER
2 THEN
2     IF .PARENT_ID EQL 0
2     THEN
3         BEGIN
3         STATUS = $CMKRNL (ROUTIN = GET_PARENT_LOCK);
3         IF NOT .STATUS THEN RETURN .STATUS;
3         END;
2
2 ! Do any initial setup for the object.  For files, this means opening the
2 ! specified file if it is not already open.  For devices, this means assigning
2 ! a channel is one is not already assigned.  For most other objects, nothing
2 ! special is needed.
2
2 CASE .LOCAL_OBJTYP FROM MIN_OBJECT_TYPE TO MAX_OBJECT_TYPE OF
2 SET
2     [ACL$C_FILE]:
3         BEGIN
3
3 ! Initialize storage.
3
3         CH$FILL (0, FIB$C_LENGTH, FILE_FIB);
```

```
  3             CH$FILL (0, DSC$C_S_BLN, FILE_FIB_DESC);
  3             FILE_FIB_DESC[DSC$W_LENGTH] = FIB$C_LENGTH;
  3             FILE_FIB_DESC[DSC$A_POINTER] = FILE_FIB;
  3             FILE_FIB[FIB$B_AGENT_MODE] = .CHANGE_ACMODE;
  3
  3   ! If the file is not accessed, do it now.
  3
  3             IF .IO_CHANNEL EQL 0
  3             THEN
  4                 BEGIN
P 4                 $FAB_INIT (FAB = FILE_FAB,
P 4                            FNS = .OBJECT_DESC[0],
P 4                            FNA = .OBJECT_DESC[1],
P 4                            FOP = UFO,
  4                            NAM = FILE_NAM);
P 4                 $NAM_INIT (NAM = FILE_NAM,
P 4                            ESA = FILE_EXP_NAME,
P 4                            ESS = NAM$C_MAXRSS,
P 4                            RSA = FILE_RES_NAME,
  4                            RSS = NAM$C_MAXRSS);
  4                 IF .SHARE
  4                 THEN
  5                     BEGIN
  5                     FILE_FAB[FAB$B_SHR] = FAB$M_GET OR FAB$M_PUT OR FAB$M_UPI;
  5                     FILE_FAB[FAB$B_FAC] = FAB$M_GET;
  5                     END
  4                 ELSE
  5                     BEGIN
  5                     FILE_FAB[FAB$B_SHR] = FAB$M_NIL;
  5                     FILE_FAB[FAB$B_FAC] = FAB$M_GET OR FAB$M_PUT;
  4                     END;
  4                 FILE_FAB[FAB$V_FILE_MODE] = .CHANGE_ACMODE;
  4
  4                 STATUS = $OPEN (FAB = FILE_FAB);
  4                 IO_CHANNEL = .FILE_FAB[FAB$L_STV];
  4                 END
  3             ELSE STATUS = SS$_NORMAL;
  3
  3   ! Now that a channel has been assigned to the file, do a simple access to
  3   ! fill the fib.  This is needed to get the file-id used to build the lock name.
  3
  3             IF .STATUS
  3             THEN
  4                 BEGIN
P 4                 STATUS = $QIOW (CHAN = .IO_CHANNEL,
P 4                                 FUNC = IO$_ACCESS,
P 4                                 IOSB = LOCAL_IOSB,
  4                                 P1 = FILE_FIB_DESC);
  4                 IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
  3                 END;
  2             END;
  2
  2         [ACL$C_DEVICE]:
  3             BEGIN
  3
  3   ! If necessary assign a channel to the specified device.
  3
```

```
     3              IF .IO_CHANNEL EQL 0
     3              THEN
     4                  BEGIN
     4                  IF .OBJECT_DESC[0] EQL 0
     4                  THEN RETURN SS$_INSFARG;
P    4                  STATUS = $ASSIGN (DEVNAM = OBJECT_DESC,
     4                                    CHAN = IO_CHANNEL);
     3                  END;
     3
     3  ! Now that there is a channel to the device, locate the ACL queue head.
     3
     3              IF .STATUS
     3              THEN
     4                  BEGIN
     4                  CMK_ARG_LIST[0] = 1;                    ! Number of args
     4                  CMK_ARG_LIST[1] = .IO_CHANNEL;          ! Channel number
P    4                  STATUS = $CMKRNL (ROUTIN = GET_UCB_ACL,
     4                                    ARGLST = CMK_ARG_LIST);
     3                  END;
     2              END;
     2
     2          [ACL$C_JOBCTL_QUEUE]:
     3              BEGIN
     3              STATUS = SS$_BADPARAM;
     2              END;
     2
     2          [ACL$C_COMMON_EF_CLUSTER]:
     3              BEGIN
     3              IF .OBJECT_DESC[0] EQL 0
     3              THEN RETURN SS$_INSFARG;
     3              CMK_ARG_LIST[0] = 1;                        ! Number of args
     3              CMK_ARG_LIST[1] = OBJECT_DESC;              ! Cluster name descr
P    3              STATUS = $CMKRNL (ROUTIN = GET_CEB_ACL,
     3                                ARGLST = CMK_ARG_LIST);
     2              END;
     2
     2          [ACL$C_LOGICAL_NAME_TABLE]:
     3              BEGIN
     3              IF .OBJECT_DESC[0] EQL 0
     3              THEN RETURN SS$_INSFARG;
     3              CMK_ARG_LIST[0] = 1;                        ! Number of args
     3              CMK_ARG_LIST[1] = OBJECT_DESC;              ! Logical name table descr
P    3              STATUS = $CMKRNL (ROUTIN = GET_LNT_ACL,
     3                                ARGLST = CMK_ARG_LIST);
     2              END;
     2
     2          [ACL$C_PROCESS]:
     3              BEGIN
     3              IF .OBJECT_DESC[0] EQL 0
     3              THEN RETURN SS$_INSFARG;
     3              CMK_ARG_LIST[0] = 1;                        ! Number of args
     3              CMK_ARG_LIST[1] = OBJECT_DESC;              ! Process name descr
P    3              STATUS = $CMKRNL (ROUTIN = GET_PCB_ACL,
     3                                ARGLST = CMK_ARG_LIST);
     2              END;
     2
     2          [ACL$C_GLOBAL_SECTION]:
```

```
  3         BEGIN
  3         IF .OBJECT_DESC[0] EQL 0
  3         THEN RETURN SS$_INSFARG;
  3         CMK_ARG_LIST[0] = 1;                              ! Number of args
  3         CMK_ARG_LIST[1] = OBJECT_DESC;                    ! Section name descr
P 3         STATUS = $CMKRNL (ROUTIN = GET_GBL_ACL,
  3                                   ARGLST = CMK_ARG_LIST);
  2         END;
  2
  2     [INRANGE, OUTRANGE]:        STATUS = SS$_BADPARAM;
  2 TES;
  2
  2 ! If any error have occurred, leave now.
  2
  2 IF NOT .STATUS
  2 THEN
  3     BEGIN
  3     IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
  3     RETURN .STATUS;
  2     END;
  2
  2 ! Now that the device has been identified, and a channel assigned if needed,
  2 ! form the remainder of the lock resource name.  Then do the appropriate lock
  2 ! or unlock.
  2
  2 IF .LOCAL_OBJTYP EQL ACL$C_FILE OR .LOCAL_OBJTYP EQL ACL$C_DEVICE
  2 THEN
  3     BEGIN
  3     LOCAL       TMP_LEN;
  3
  3 ! Build the remaining portion of the lock name.
  3
  3     DVI_ATR_LIST[0, ITM$W_ITMCOD] = DVI$_DEVLOCKNAM;
  3     DVI_ATR_LIST[0, ITM$W_BUFSIZ] = 31 - RSN_S_PREFIX;
  3     DVI_ATR_LIST[0, ITM$L_BUFADR] = RESNAM_TEXT[RSN_T_DEVNAM];
  3     DVI_ATR_LIST[0, ITM$L_RETLEN] = TMP_LEN;
P 3     STATUS = $GETDVI (CHAN = .IO_CHANNEL,
P 3                       ITMLST = DVI_ATR_LIST,
  3                       IOSB = LOCAL_IOSB);
  3     IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
  3     IF NOT .STATUS
  3     THEN
  4         BEGIN
  4         IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
  4         RETURN .STATUS;
  3         END;
  3
  3     LOCK_RESNAM[DSC$W_LENGTH] = .LOCK_RESNAM[DSC$W_LENGTH] + .TMP_LEN;
  3     IF .LOCAL_OBJTYP EQL ACL$C_FILE
  3     THEN
  4         BEGIN
  4         RESNAM_TEXT[RSN_W_FID_NUM] = .FILE_FIB[FIB$W_FID_NUM];
  4         RESNAM_TEXT[RSN_W_FID_SEQ] = .FILE_FIB[FIB$W_FID_SEQ];
  4         LOCK_RESNAM[DSC$W_LENGTH] = .LOCK_RESNAM[DSC$W_LENGTH] + 4;
  3         END;
  2     END;
  2
```

```
! For files, process the attribute list, and pass it through to the ACP.
! for all other objects, the attribute processing is done here.

IF .LOCAL_OBJTYP EQL ACL$C_FILE
THEN
    BEGIN

! Build the ACP attribute list.

    STATUS = LIB$GET_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
    IF NOT .STATUS
    THEN
        BEGIN
        IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
        RETURN .STATUS;
        END;
    FUNCTION_CODE = IO$_ACCESS;
    ACP_ATR_PTR = 0;
    INCR J FROM 0 TO .ITEM_COUNT - 1
    DO
        BEGIN
        IF PROBER (%REF (0), %REF (ITM$S_ITEM), ITEM_LIST[.J, 0,0,0,0])
        THEN
            BEGIN
            ITEM_CODE = .ITEM_LIST[.J, ITM$W_ITMCOD];
            ITEM_SIZE = .ITEM_LIST[.J, ITM$W_BUFSIZ];
            ITEM_ADDR = .ITEM_LIST[.J, ITM$L_BUFADR];
            END
        ELSE
            BEGIN
            STATUS = SS$_ACCVIO;
            LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
            IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
            RETURN .STATUS;
            END;
        IF .ITEM_CODE GTR MAX_ACL_ATR
        THEN
            BEGIN
            STATUS = SS$_BADPARAM;
            LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
            IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
            RETURN .STATUS;
            END;

        IF .ITEM_CODE EQL ACL$C_RLOCK_ACL
        OR .ITEM_CODE EQL ACL$C_WLOCK_ACL
        THEN
            BEGIN
            IF .ITEM_SIZE LSSU 4
            THEN
                BEGIN
                STATUS = SS$_BADPARAM;
                LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
                IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
                RETURN .STATUS;
                END;
            IF NOT EXE$PROBEW (0, .ITEM_SIZE, .ITEM_ADDR)
```

```
  5                     THEN
  6                          BEGIN
  6                          STATUS = SS$_ACCVIO;
  6                          LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
  6                          IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
  6                          RETURN .STATUS;
  5                          END;
P 5                     STATUS = $ENQ (LKMODE = (IF .ITEM_CODE EQL ACL$C_RLOCK_ACL
P 5                                                  THEN LCK$K_CRMODE ELSE LCK$K_PWMODE),
P 5                                    LKSB = LOCAL_IOSB,
P 5                                    RESNAM = LOCK_RESNAM,
P 5                                    PARID = (IF .CALL_ACMODE EQL PSL$C_USER
P 5                                                 THEN .PARENT_ID
P 5                                                 ELSE 0),
P 5                                    FLAGS = LCK$M_NOQUEUE OR
P 5                                            LCK$M_SYNCSTS OR
P 5                                            LCK$M_SYSTEM,
  5                                    ACMODE = PSL$C_USER);
  5                     IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
  5                     IF NOT .STATUS
  5                     THEN
  6                          BEGIN
  6                          LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
  6                          IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
  6                          RETURN .STATUS;
  5                          END;
  5                     CH$COPY (4, LOCAL_IOSB[2],
  5                              0,
  5                              .ITEM_SIZE,  .ITEM_ADDR);  ! Copy lock-id
  5                     END
  5
  4                 ELSE IF .ITEM_CODE EQL ACL$C_UNLOCK_ACL
  4                 THEN
  5                     BEGIN
  5                     IF .ITEM_SIZE LSSU 4
  5                     THEN
  6                          BEGIN
  6                          STATUS = SS$_BADPARAM;
  6                          LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
  6                          IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
  6                          RETURN .STATUS;
  5                          END;
  5                     IF NOT EXE$PROBER (0, .ITEM_SIZE, .ITEM_ADDR)
  5                     THEN
  6                          BEGIN
  6                          STATUS = SS$_ACCVIO;
  6                          LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
  6                          IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);
  6                          RETURN .STATUS;
  5                          END;
  5                     CH$COPY (.ITEM_SIZE, .ITEM_ADDR, 0, 4, LOCAL_LOCKID);
  5                     END
  4                 ELSE
  5                     BEGIN
  5
  5   ! Save the converted attribute code and other information.
  5
```

```
                        ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_TYPE] = .ACL_TO_ATR_TAB[.ITEM_CODE];
                        IF .ITEM_CODE EQL ACL$C_ADDACLENT OR .ITEM_CODE EQL ACL$C_DELACLENT
                        OR .ITEM_CODE EQL ACL$C_MODACLENT OR .ITEM_CODE EQL ACL$C_DELETEACL
                        THEN FUNCTION_CODE = IO$_MODIFY;
                        ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_SIZE] = .ITEM_SIZE;
                        ACP_ATR_LIST[.ACP_ATR_PTR, ATR$L_ADDR] = .ITEM_ADDR;
                        ACP_ATR_PTR = .ACP_ATR_PTR + 1;
                        END;
                END;

        ! Tie off the attribute descriptor list.

            ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_TYPE] = 0;
            ACP_ATR_LIST[.ACP_ATR_PTR, ATR$W_SIZE] = 0;

        ! Initialize the FIB, and call the ACP to process the attribute list.

            FILE_FIB[FIB$L_ACLCTX] = .ACL_CONTEXT;
            STATUS = $QIOW (CHAN = .IO_CHANNEL,
                            FUNC = .FUNCTION_CODE,
                            IOSB = LOCAL_IOSB,
                            P1 = FILE_FIB_DESC,
                            P5 = .ACP_ATR_LIST);
            IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
            IF .STATUS THEN STATUS = .FILE_FIB[FIB$L_ACL_STATUS];
            IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);

            STATUS2 = LIB$FREE_VM (%REF ((.ITEM_COUNT + 1) * 8), ACP_ATR_LIST);
            IF .STATUS AND NOT .STATUS2 THEN STATUS = .STATUS2;

        ! If an unlock request was made on the file's ACL, do it now.

            IF .LOCAL_LOCKID NEQ 0 THEN STATUS = $DEQ (LKID = .LOCAL_LOCKID);
            END

        ! For non-file objects, the queue head has been located; loop through
        ! the item list performing the actions specified.

        ELSE
            BEGIN
            CMK_ARG_LIST[0] = 3;
            CMK_ARG_LIST[1] = .ITEM_COUNT;
            CMK_ARG_LIST[2] = .ITEM_LIST;
            CMK_ARG_LIST[3] = .SHARE;
            STATUS = $CMKRNL (ROUTIN = ACL_DISPATCH,
                              ARGLST = CMK_ARG_LIST);
            END;

        ! If necessary, deassign the channel assigned.

        IF .LOCAL_CHANNEL EQL 0 THEN $DASSGN (CHAN = .IO_CHANNEL);

        ! If necessary, return the context.

        IF .CONTEXT NEQA 0
        THEN IF PROBEW (%REF (0), %REF (4), .CONTEXT)
            THEN
```

```
; 1965          1960  3           BEGIN
; 1966          1961  3           IF .LOCAL_OBJTYP EQL ACL$C_FILE
; 1967          1962  3           THEN .CONTEXT = .FILE_FIB[FIB$L_ACLCTX]
; 1968          1963  3           ELSE .CONTEXT = .ACL_CONTEXT;
; 1969          1964  3           END
; 1970          1965  2       ELSE STATUS = SS$_ACCVIO;
; 1971          1966  2
; 1972          1967  2  RETURN .STATUS;
; 1973          1968  2
; 1974          1969  1  END;                                                       ! End of routine SYS$CHANGE_ACL
; INFO#250             L1:1779
; Referenced LOCAL symbol TMP_LEN is probably not initialized



                                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

00000023  00000022  00000021  00000020  0000001F  00000000  0054C P.AFL:   .LONG   0, 31, 32, 33, 34, 35, 36, 37, 38, 39        ;
                    00000027  00000026  00000025  00000024  00564                                                                ;

                                                                   .EXTRN  SYS$CMKRNL, SYS$OPEN
                                                                   .EXTRN  SYS$QIOW, SYS$ASSIGN
                                                                   .EXTRN  SYS$DASSGN, SYS$GETDVI
                                                                   .EXTRN  SYS$ENQ, SYS$DEQ

                                                                   .PSECT  $CODE$,NOWRT,2

                                                    0FFC 00000     .ENTRY  SYS$CHANGE_ACL, Save R2,R3,R4,R5,R6,R7,R8,-  ; 1391
                                                                           R9,R10,R11
                                5E      FC68    CE   9E 00002      MOVAB   -920(SP), SP                                 ;
           34         00 00000000'  EF          28   2C 00007      MOVC5   #40, P.AFL, #0, #52, ACL_TO_ATR_TAB          ; 1486
                                          44    AE      00010                                                           ;
                                    00000000'   EF   D4 00012      CLRL    CHANGE_ACMODE                                ; 1493
                                50        14    AC   D0 00018      MOVL    ACCESS_MODE, R0                              ; 1494
                                                50   D5 0001C      TSTL    R0                                           ;
                                                0D   13 0001E      BEQL    1$                                           ;
                      60        04              00   0C 00020      PROBER  #0, #4, (R0)                                 ; 1495
                                                6A   13 00024      BEQL    5$                                           ;
                         00000000'  EF          60   D0 00026      MOVL    (R0), CHANGE_ACMODE                          ; 1496
                                                50   DC 0002D 1$:  MOVPSL  PSL                                          ; 1499
00000000'  EF         50        02              16   EF 0002F      EXTZV   #22, #2, PSL, CALL_ACMODE                    ; 1500
                                50 00000000'    EF   D0 00038      MOVL    CHANGE_ACMODE, R0                            ; 1501
                         00000000'  EF          50   D1 0003F      CMPL    R0, CALL_ACMODE                              ;
                                                07   1E 00046      BGEQU   2$                                           ;
                                50 00000000'    EF   D0 00048      MOVL    CALL_ACMODE, R0                              ;
                         00000000'  EF          50   D0 0004F 2$:  MOVL    R0, CHANGE_ACMODE                            ;
                                03 00000000'    EF   D1 00056      CMPL    CHANGE_ACMODE, #3                            ; 1505
                                                3D   1A 0005D      BGTRU   6$                                           ;
                                50        04    AC   3C 0005F      MOVZWL  CHANNEL, R0                                  ; 1509
                             10 AE              50   B0 00063      MOVW    R0, LOCAL_CHANNEL                            ;
                             20 AE              50   B0 00067      MOVW    R0, IO_CHANNEL                               ;
                                58        20    AE   3C 0006B      MOVZWL  IO_CHANNEL, R8                               ; 1510
                                                12   13 0006F      BEQL    3$                                           ;
                                50              58   D0 00071      MOVL    R8, R0                                       ; 1513
                                   00000000G    00   16 00074      JSB     IOC$VERIFYCHAN                               ;
                                5B              50   D0 0007A      MOVL    R0, STATUS                                   ;
                                03              5B   E8 0007D      BLBS    STATUS, 3$                                   ; 1514
```

```
                                   059B  31 00080           BRW     81$                                       ;
                    50       08      AC  D0 00083 3$:       MOVL    OBJECT_TYPE, R0                           ; 1519
                                     03  12 00087           BNEQ    4$                                        ;
                                   0293  31 00089           BRW     36$                                       ;
          60        04               00  0C 0008C 4$:       PROBER  #0, #4, (R0)                              ; 1520
                                     6D  13 00090 5$:       BEQL    12$                                       ;
                    56               60  D0 00092           MOVL    (R0), LOCAL_OBJTYP                        ; 1521
                                     05  13 00095           BEQL    6$                                        ; 1527
                    07               56  D1 00097           CMPL    LOCAL_OBJTYP, #7                          ; 1528
                                     04  1B 0009A           BLEQU   7$                                        ;
                    50               14  D0 0009C 6$:       MOVL    #20, R0                                   ; 1529
                                         04 0009F           RET                                               ;
                    50       0C      AC  D0 000A0 7$:       MOVL    OBJECT_NAME, R0                           ; 1533
                                     24  13 000A4           BEQL    8$                                        ;
          60        08               00  0C 000A6           PROBER  #0, #8, (R0)                              ; 1536
                                     53  13 000AA           BEQL    12$                                       ;
               F0   AD               60  3C 000AC           MOVZWL  (R0), OBJECT_DESC                         ; 1538
               F4   AD       04      A0  D0 000B0           MOVL    4(R0), OBJECT_DESC+4                      ; 1539
                    50       F4      AD  D0 000B5           MOVL    OBJECT_DESC+4, R0                         ; 1540
                    51       F0      AD  D0 000B9           MOVL    OBJECT_DESC, R1                           ;
                                     53  D4 000BD           CLRL    R3                                        ;
                       00000000G     00  16 000BF           JSB     EXE$PROBER                                ;
                    05               50  E8 000C5           BLBS    R0, 9$                                    ;
                                     62  11 000C8           BRB     15$                                       ; 1541
                             F0      AD  7C 000CA 8$:       CLRQ    OBJECT_DESC                               ; 1545
                       00000000'     EF  D4 000CD 9$:       CLRL    ACL_CONTEXT                               ; 1551
                             1C      AE  D4 000D3           CLRL    28(SP)                                    ; 1552
                             1C      AC  D5 000D6           TSTL    CONTEXT                                   ;
                                     12  13 000D9           BEQL    10$                                       ;
                             1C      AE  D6 000DB           INCL    28(SP)                                    ;
     1C   BC        04               00  0D 000DE           PROBEW  #0, #4, @CONTEXT                          ; 1553
                                     47  13 000E3           BEQL    15$                                       ;
           00000000'   EF    1C      BC  D0 000E5           MOVL    @CONTEXT, ACL_CONTEXT                     ; 1554
                       5A            01  90 000ED 10$:      MOVB    #1, SHARE                                 ; 1559
                       59    10      AC  D0 000F0           MOVL    ITEM_LIST, R9                             ; 1561
                                     50  D4 000F4           CLRL    J                                         ;
          51        50               0C  C5 000F6 11$:      MULL3   #12, J, R1                                ;
        6149        0C               00  0C 000FA           PROBER  #0, #12, (R1)[R9]                         ;
                                     2B  13 000FF 12$:      BEQL    15$                                       ;
                                   6149  9F 00101           PUSHAB  (R1)[R9]                                  ; 1562
                                     9E  B5 00104           TSTW    @(SP)+                                    ;
                                     05  12 00106           BNEQ    13$                                       ;
                    57               50  D0 00108           MOVL    J, ITEM_COUNT                             ; 1565
                                     2B  11 0010B           BRB     17$                                       ; 1564
                                02 A149  9F 0010D 13$:      PUSHAB  2(R1)[R9]                                 ; 1570
                    51               9E  3C 00111           MOVZWL  @(SP)+, R1                                ;
                    01               51  B1 00114           CMPW    R1, #1                                    ;
                                     0F  13 00117           BEQL    14$                                       ;
                    02               51  B1 00119           CMPW    R1, #2                                    ; 1571
                                     0A  13 0011C           BEQL    14$                                       ;
                    03               51  B1 0011E           CMPW    R1, #3                                    ; 1572
                                     05  13 00121           BEQL    14$                                       ;
                    06               51  B1 00123           CMPW    R1, #6                                    ; 1573
                                     08  12 00126           BNEQ    16$                                       ;
                                     5A  94 00128 14$:      CLRB    SHARE                                     ; 1574
                                     04  11 0012A           BRB     16$                                       ; 1562
                    50               0C  D0 0012C 15$:      MOVL    #12, R0                                   ; 1576
```

```
                                          04 0012F        RET                                                  ;
                 BE        50 7FFFFFFF    8F F3 00130 16$: AOBLEQ  #2147483647, J, 11$                         ; 1561
      18         00        6E             00 2C 00138 17$: MOVC5   #0, (SP), #0, #24, DVI_ATR_LIST             ; 1580
                                  78      AE    0013D                                                          ;
      08         00        6E             00 2C 0013F     MOVC5   #0, (SP), #0, #8, LOCK_RESNAM                ; 1581
                           00000000'      EF    00144                                                          ;
                                  28      AE D4 00149     CLRL    LOCAL_LOCKID                                 ; 1582
              00000000'    EF             08 B0 0014C     MOVW    #8, LOCK_RESNAM                              ; 1586
              00000000'    EF 00000000'   EF 9E 00153     MOVAB   RESNAM_TEXT, LOCK_RESNAM+4                   ; 1587
                           50 00000000'EF46 D0 0015E      MOVL    LOCK_PREFIX[LOCAL_OBJTYP], R0                ; 1588
      08         00        04             B0             
                                          60 2C 00166     MOVC5   (R0), @4(R0), #0, #8, RESNAM_TEXT            ;
                           00000000'      EF    0016C                                                          ;
                        03 00000000'      EF D1 00171     CMPL    CALL_ACMODE, #3                              ; 1596
                                          20 12 00178     BNEQ    18$                                          ;
                           00000000'      EF D5 0017A     TSTL    PARENT_ID                                    ; 1598
                                          18 12 00180     BNEQ    18$                                          ;
                                          7E D4 00182     CLRL    -(SP)                                        ; 1601
                           00000000V      EF 9F 00184     PUSHAB  GET_PARENT_LOCK                              ;
              00000000G    00             02 FB 0018A     CALLS   #2, SYS$CMKRNL                               ;
                           5B             50 D0 00191     MOVL    R0, STATUS                                   ;
                           03             5B E8 00194     BLBS    STATUS, 18$                                  ; 1602
                                        0484 31 00197     BRW     81$                                          ;
                 06        01             56 CF 0019A 18$: CASEL   LOCAL_OBJTYP, #1, #6                        ; 1610
    0131       012C      00F7           0011    0019E 19$: .WORD   20$-19$,-                                   ;
               017C      0163           014A    001A6             26$-19$,-                                    ;
                                                                  29$-19$,-                                    ;
                                                                  31$-19$,-                                    ;
                                                                  32$-19$,-                                    ;
                                                                  33$-19$,-                                    ;
                                                                  35$-19$                                      ;
                                        011B 31 001AC     BRW     29$                                          ; 1741
    0040  8F         00          6E       00 2C 001AF 20$: MOVC5   #0, (SP), #0, #64, FILE_FIB                 ; 1617
                              0090        CE    001B6                                                          ;
          08         00          6E       00 2C 001B9     MOVC5   #0, (SP), #0, #8, FILE_FIB_DESC              ; 1618
                              00D0        CE    001BE                                                          ;
               00D0  CE          40       8F 9B 001C1     MOVZBW  #64, FILE_FIB_DESC                           ; 1619
               00D4  CE        0090       CE 9E 001C7     MOVAB   FILE_FIB, FILE_FIB_DESC+4                    ; 1620
               00BE  CE 00000000'         EF 90 001CE     MOVB    CHANGE_ACMODE, FILE_FIB+46                   ; 1621
                                          58 D5 001D7     TSTL    R8                                           ; 1625
                                          03 13 001D9     BEQL    21$                                          ;
                                        0087 31 001DB     BRW     24$                                          ;
    0050  8F         00          6E       00 2C 001DE 21$: MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 1632
                                  A0      AD    001E5                                                          ;
                 A0  AD        5003       8F B0 001E7     MOVW    #20483, $RMS_PTR                             ;
                 A4  AD    00020000       8F D0 001ED     MOVL    #131072, $RMS_PTR+4                          ;
                 B6  AD                   02 90 001F5     MOVB    #2, $RMS_PTR+22                              ;
                 BF  AD                   02 90 001F9     MOVB    #2, $RMS_PTR+31                              ;
                 C8  AD        FF40       CD 9E 001FD     MOVAB   FILE_NAM, $RMS_PTR+40                        ;
                 CC  AD          F4       AD D0 00203     MOVL    OBJECT_DESC+4, $RMS_PTR+44                   ;
                 D4  AD          F0       AD 90 00208     MOVB    OBJECT_DESC, $RMS_PTR+52                     ;
    0060  8F         00          6E       00 2C 0020D     MOVC5   #0, (SP), #0, #96, $RMS_PTR                  ; 1637
                               FF40       CD    00214                                                          ;
               FF40  CD        6002       8F B0 00217     MOVW    #24578, $RMS_PTR                             ;
               FF42  CD          01       8E    0021E     MNEGB   #1, $RMS_PTR+2                               ;
               FF44  CD        00D8       CE 9E 00223     MOVAB   FILE_RES_NAME, $RMS_PTR+4                    ;
               FF4A  CD          01       8E    0022A     MNEGB   #1, $RMS_PTR+10                              ;
               FF4C  CD        FE40       CD 9E 0022F     MOVAB   FILE_EXP_NAME, $RMS_PTR+12                   ;
```

```
                         08            5A  E9 00236       BLBC    SHARE, 22$                          ; 1638
                    B6   AD      4302  8F  B0 00239       MOVW    #17154, FILE_FAB+22                 ; 1642
                         06            11     0023F       BRB     23$                                 ; 1638
                    B6   AD      2003  8F  B0 00241 22$:  MOVW    #8195, FILE_FAB+22                  ; 1647
 EA   AD        02       04 00000000'  EF  F0 00247 23$:  INSV    CHANGE_ACMODE, #4, #2, FILE_FAB+74  ; 1649
                                 A0    AD  9F 00251       PUSHAB  FILE_FAB                            ; 1651
              00000000G  00            01  FB 00254       CALLS   #1, SYS$OPEN                        ;
                         5B            50  D0 0025B       MOVL    R0, STATUS                          ;
                    20   AE      AC    AD  B0 0025E       MOVW    FILE_FAB+12, IO_CHANNEL             ; 1652
                         03            11     00263       BRB     25$                                 ; 1625
                         5B            01  D0 00265 24$:  MOVL    #1, STATUS                          ; 1654
                         45            5B  E9 00268 25$:  BLBC    STATUS, 27$                         ; 1659
                                       7E  7C 0026B       CLRQ    -(SP)                               ; 1665
                                       7E  7C 0026D       CLRQ    -(SP)                               ;
                                       7E  D4 0026F       CLRL    -(SP)                               ;
                               00E4    CE  9F 00271       PUSHAB  FILE_FIB_DESC                       ;
                                       7E  7C 00275       CLRQ    -(SP)                               ;
                                 F8    AD  9F 00277       PUSHAB  LOCAL_IOSB                          ;
                                       32  DD 0027A       PUSHL   #50                                 ;
                         7E      48    AE  3C 0027C       MOVZWL  IO_CHANNEL, -(SP)                   ;
                                       7E  D4 00280       CLRL    -(SP)                               ;
              00000000G  00            0C  FB 00282       CALLS   #12, SYS$QIOW                       ;
                         5B            50  D0 00289       MOVL    R0, STATUS                          ;
                         21            5B  E9 0028C       BLBC    STATUS, 27$                         ; 1666
                         5B      F8    AD  3C 0028F       MOVZWL  LOCAL_IOSB, STATUS                  ;
                                       38  11 00293       BRB     30$                                 ; 1610
                                       58  D5 00295 26$:  TSTL    R8                                  ; 1675
                                       17  12 00297       BNEQ    27$                                 ;
                                 F0    AD  D5 00299       TSTL    OBJECT_DESC                         ; 1678
                                       66  13 0029C       BEQL    34$                                 ;
                                       7E  7C 0029E       CLRQ    -(SP)                               ; 1681
                                 28    AE  9F 002A0       PUSHAB  IO_CHANNEL                          ;
                                 F0    AD  9F 002A3       PUSHAB  OBJECT_DESC                         ;
              00000000G  00            04  FB 002A6       CALLS   #4, SYS$ASSIGN                      ;
                         5B            50  D0 002AD       MOVL    R0, STATUS                          ;
                         03            5B  E8 002B0 27$:  BLBS    STATUS, 28$                         ; 1686
                                   01F9    31 002B3       BRW     60$                                 ;
                    30   AE            01  D0 002B6 28$:  MOVL    #1, CMK_ARG_LIST                    ; 1689
                    34   AE      20    AE  3C 002BA       MOVZWL  IO_CHANNEL, CMK_ARG_LIST+4          ; 1690
                                 30    AE  9F 002BF       PUSHAB  CMK_ARG_LIST                        ; 1692
                          00000000V    EF  9F 002C2       PUSHAB  GET_UCB_ACL                         ;
                                       6D  11 002C8       BRB     38$                                 ;
                         5B            14  D0 002CA 29$:  MOVL    #20, STATUS                         ; 1698
                                       72  11 002CD 30$:  BRB     39$                                 ; 1610
                                 F0    AD  D5 002CF 31$:  TSTL    OBJECT_DESC                         ; 1703
                                       4B  13 002D2       BEQL    36$                                 ;
                    30   AE            01  D0 002D4       MOVL    #1, CMK_ARG_LIST                    ; 1705
                    34   AE      F0    AD  9E 002D8       MOVAB   OBJECT_DESC, CMK_ARG_LIST+4         ; 1706
                                 30    AE  9F 002DD       PUSHAB  CMK_ARG_LIST                        ; 1708
                          00000000V    EF  9F 002E0       PUSHAB  GET_CEB_ACL                         ;
                                       4F  11 002E6       BRB     38$                                 ;
                                 F0    AD  D5 002E8 32$:  TSTL    OBJECT_DESC                         ; 1713
                                       32  13 002EB       BEQL    36$                                 ;
                    30   AE            01  D0 002ED       MOVL    #1, CMK_ARG_LIST                    ; 1715
                    34   AE      F0    AD  9E 002F1       MOVAB   OBJECT_DESC, CMK_ARG_LIST+4         ; 1716
                                 30    AE  9F 002F6       PUSHAB  CMK_ARG_LIST                        ; 1718
                          00000000V    EF  9F 002F9       PUSHAB  GET_LNT_ACL                         ;
```

```
                                        36  11 002FF          BRB     38$                                          ;
                                   F0   AD  D5 00301 33$:     TSTL    OBJECT_DESC                                  ; 1723
                                        19  13 00304 34$:     BEQL    36$                                          ;
                  30   AE               01  D0 00306          MOVL    #1, CMK_ARG_LIST                             ; 1725
                  34   AE          F0   AD  9E 0030A          MOVAB   OBJECT_DESC, CMK_ARG_LIST+4                  ; 1726
                                   30   AE  9F 0030F          PUSHAB  CMK_ARG_LIST                                 ; 1728
                           00000000V    EF  9F 00312          PUSHAB  GET_PCB_ACL                                  ;
                                        1D  11 00318          BRB     38$                                          ;
                                   F0   AD  D5 0031A 35$:     TSTL    OBJECT_DESC                                  ; 1733
                                        06  12 0031D          BNEQ    37$                                          ;
                       50        0114   8F  3C 0031F 36$:     MOVZWL  #276, R0                                     ; 1734
                                            04 00324          RET                                                  ;
                  30   AE               01  D0 00325 37$:     MOVL    #1, CMK_ARG_LIST                             ; 1735
                  34   AE          F0   AD  9E 00329          MOVAB   OBJECT_DESC, CMK_ARG_LIST+4                  ; 1736
                                   30   AE  9F 0032E          PUSHAB  CMK_ARG_LIST                                 ; 1738
                           00000000V    EF  9F 00331          PUSHAB  GET_GBL_ACL                                  ;
             00000000G  00              02  FB 00337 38$:     CALLS   #2, SYS$CMKRNL                               ;
                        5B              50  D0 0033E          MOVL    R0, STATUS                                   ;
                        4C              5B  E9 00341 39$:     BLBC    STATUS, 42$                                  ; 1746
                                   18   AE  D4 00344          CLRL    24(SP)                                       ; 1757
                        01              56  D1 00347          CMPL    LOCAL_OBJTYP, #1                             ;
                                        05  12 0034A          BNEQ    40$                                          ;
                                   18   AE  D6 0034C          INCL    24(SP)                                       ;
                                        05  11 0034F          BRB     41$                                          ;
                        02              56  D1 00351 40$:     CMPL    LOCAL_OBJTYP, #2                             ;
                                        59  12 00354          BNEQ    43$                                          ;
                  78   AE 00F00017      8F  D0 00356 41$:     MOVL    #15728663, DVI_ATR_LIST                      ; 1765
                  7C   AE 00000000'     EF  9E 0035E          MOVAB   RESNAM_TEXT+8, DVI_ATR_LIST+4                ; 1766
                0080   CE          24   AE  9E 00366          MOVAB   TMP_LEN, DVI_ATR_LIST+8                      ; 1767
                                        7E  7C 0036C          CLRQ    -(SP)                                        ; 1770
                                        7E  D4 0036E          CLRL    -(SP)                                        ;
                                   F8   AD  9F 00370          PUSHAB  LOCAL_IOSB                                   ;
                                 0088   CE  9F 00373          PUSHAB  DVI_ATR_LIST                                 ;
                                        7E  D4 00377          CLRL    -(SP)                                        ;
                        7E         38   AE  3C 00379          MOVZWL  IO_CHANNEL, -(SP)                            ;
                                        7E  D4 0037D          CLRL    -(SP)                                        ;
             00000000G  00              08  FB 0037F          CALLS   #8, SYS$GETDVI                               ;
                        5B              50  D0 00386          MOVL    R0, STATUS                                   ;
                        48              5B  E9 00389          BLBC    STATUS, 45$                                  ; 1771
                        5B         F8   AD  3C 0038C          MOVZWL  LOCAL_IOSB, STATUS                           ;
                        41              5B  E9 00390 42$:     BLBC    STATUS, 45$                                  ; 1772
             00000000'  EF         24   AE  A0 00393          ADDW2   TMP_LEN, LOCK_RESNAM                         ; 1779
                        10         18   AE  E9 0039B          BLBC    24(SP), 43$                                  ; 1780
             00000000'  EF       0094   CE  D0 0039F          MOVL    FILE_FIB+4, RESNAM_TEXT+24                   ; 1783
             00000000'  EF              04  A0 003A8          ADDW2   #4, LOCK_RESNAM                              ; 1785
                        03         18   AE  E8 003AF 43$:     BLBS    24(SP), 44$                                  ; 1798
                                 0211   31 003B3              BRW     75$                                          ;
                                   2C   AE  9F 003B6 44$:     PUSHAB  ACP_ATR_LIST                                 ;
         08   AE                   57   03  78 003B9          ASHL    #3, ITEM_COUNT, 8(SP)                        ;
                  08   AE               08  C0 003BE          ADDL2   #8, 8(SP)                                    ;
                  18   AE          08   AE  D0 003C2          MOVL    8(SP), 24(SP)                                ;
                                   18   AE  9F 003C7          PUSHAB  24(SP)                                       ;
             00000000G  00              02  FB 003CA          CALLS   #2, LIB$GET_VM                               ;
                        5B              50  D0 003D1          MOVL    R0, STATUS                                   ;
                        03              5B  E8 003D4 45$:     BLBS    STATUS, 46$                                  ; 1799
                                 00D5   31 003D7              BRW     60$                                          ;
                  14   AE               32  D0 003DA 46$:     MOVL    #50, FUNCTION_CODE                           ; 1805
```

```
                                           58  D4 003DE          CLRL    ACP_ATR_PTR                                  ; 1806
                          56               01  CE 003E0          MNEGL   #1, J                                        ; 1807
                                         014D  31 003E3          BRW     69$
               50         56               0C  C5 003E6 47$:     MULL3   #12, J, R0                                   ; 1810
             6049         0C               00  0C 003EA          PROBER  #0, #12, (R0)[R9]
                                           18  13 003EF          BEQL    48$
                                      02 A049  9F 003F1          PUSHAB  2(R0)[R9]                                    ; 1813
                          5A               9E  3C 003F5          MOVZWL  @(SP)+, ITEM_CODE
                                         6049  9F 003F8          PUSHAB  (R0)[R9]                                     ; 1814
                     08   AE               9E  3C 003FB          MOVZWL  @(SP)+, ITEM_SIZE
                                      04 A049  9F 003FF          PUSHAB  4(R0)[R9]                                    ; 1815
                     0C   AE               9E  D0 00403          MOVL    @(SP)+, ITEM_ADDR
                                           06  11 00407          BRB     49$                                          ; 1810
                          5B               0C  D0 00409 48$:     MOVL    #12, STATUS                                  ; 1819
                                         008E  31 0040C          BRW     59$                                          ; 1820
                          0C               5A  D1 0040F 49$:     CMPL    ITEM_CODE, #12                               ; 1824
                                           03  15 00412          BLEQ    50$
                                         0083  31 00414          BRW     58$
                                           54  D4 00417 50$:     CLRL    R4                                           ; 1833
                          0A               5A  D1 00419          CMPL    ITEM_CODE, #10
                                           04  12 0041C          BNEQ    51$
                                           54  D6 0041E          INCL    R4
                                           05  11 00420          BRB     52$
                          0B               5A  D1 00422 51$:     CMPL    ITEM_CODE, #11                               ; 1834
                                           68  12 00425          BNEQ    57$
                          04          08   AE  D1 00427 52$:     CMPL    ITEM_SIZE, #4                                ; 1837
                                           6D  1F 0042B          BLSSU   58$
                          50          0C   AE  D0 0042D          MOVL    ITEM_ADDR, R0                                ; 1845
                          51          08   AE  D0 00431          MOVL    ITEM_SIZE, R1
                                           53  D4 00435          CLRL    R3
                  00000000G                00  16 00437          JSB     EXE$PROBEW
                          C9               50  E9 0043D          BLBC    R0, 48$
                          7E               03  7D 00440          MOVQ    #3, -(SP)                                    ; 1863
                                           7E  7C 00443          CLRQ    -(SP)
                                           7E  D4 00445          CLRL    -(SP)
                       03 00000000'        EF  D1 00447          CMPL    CALL_ACMODE, #3
                                           08  12 0044E          BNEQ    53$
                          00000000'        EF  DD 00450          PUSHL   PARENT_ID
                                           02  11 00456          BRB     54$
                                           7E  D4 00458 53$:     CLRL    -(SP)
                          00000000'        EF  9F 0045A 54$:     PUSHAB  LOCK_RESNAM
                                           1C  DD 00460          PUSHL   #28
                                      F8   AD  9F 00462          PUSHAB  LOCAL_IOSB
                          04               54  E9 00465          BLBC    R4, 55$
                                           01  DD 00468          PUSHL   #1
                                           02  11 0046A          BRB     56$
                                           04  DD 0046C 55$:     PUSHL   #4
                                           7E  D4 0046E 56$:     CLRL    -(SP)
                00000000G 00               0B  FB 00470          CALLS   #11, SYS$ENQ
                          5B               50  D0 00477          MOVL    R0, STATUS
                          20               5B  E9 0047A          BLBC    STATUS, 59$                                  ; 1864
                          5B          F8   AD  3C 0047D          MOVZWL  LOCAL_IOSB, STATUS
                          19               5B  E9 00481          BLBC    STATUS, 59$                                  ; 1865
   08   AE          00          FC         AD  04  2C 00484     MOVC5   #4, LOCAL_IOSB+4, #0, ITEM_SIZE, @ITEM_ADDR   ; 1874
                                      0C   BE     0048B
                                           6B  11 0048D          BRB     65$                                          ; 1833
                          0C               5A  D1 0048F 57$:     CMPL    ITEM_CODE, #12                               ; 1877
```

```
                                         68  12 00492         BNEQ    66$                                              ;
                     04          08      AE  D1 00494         CMPL    ITEM_SIZE, #4                                    ; 1880
                                         1C  1E 00498         BGEQU   61$                                              ;
                     5B                  14  D0 0049A  58$:   MOVL    #20, STATUS                                      ; 1883
                                 2C      AE  9F 0049D  59$:   PUSHAB  ACP_ATR_LIST                                     ; 1884
                04   AE          08      AE  D0 004A0         MOVL    8(SP), 4(SP)                                     ;
                                 04      AE  9F 004A5         PUSHAB  4(SP)                                            ;
           00000000G 00                  02  FB 004A8         CALLS   #2, LIB$FREE_VM                                  ;
                                 10      AE  B5 004AF  60$:   TSTW    LOCAL_CHANNEL                                    ; 1885
                                         2F  13 004B2         BEQL    62$                                              ;
                                         38  11 004B4         BRB     63$                                              ; 1886
                     50          0C      AE  D0 004B6  61$:   MOVL    ITEM_ADDR, R0                                    ; 1888
                     51          08      AE  D0 004BA         MOVL    ITEM_SIZE, R1                                    ;
                                         53  D4 004BE         CLRL    R3                                               ;
                          00000000G      00  16 004C0         JSB     EXE$PROBER                                       ;
                                 28      50  E8 004C6         BLBS    R0, 64$                                          ;
                                 5B      0C  D0 004C9         MOVL    #12, STATUS                                      ; 1891
                                 2C      AE  9F 004CC         PUSHAB  ACP_ATR_LIST                                     ; 1892
                04   AE          08      AE  D0 004CF         MOVL    8(SP), 4(SP)                                     ;
                                 04      AE  9F 004D4         PUSHAB  4(SP)                                            ;
           00000000G 00                  02  FB 004D7         CALLS   #2, LIB$FREE_VM                                  ;
                                 10      AE  B5 004DE         TSTW    LOCAL_CHANNEL                                    ; 1893
                                         0B  12 004E1         BNEQ    63$                                              ;
                     7E          20      AE  3C 004E3  62$:   MOVZWL  IO_CHANNEL, -(SP)                                ;
           00000000G 00                  01  FB 004E7         CALLS   #1, SYS$DASSGN                                   ;
                                       012D  31 004EE  63$:   BRW     81$                                              ; 1894
  04            00   0C   BE     08      AE  2C 004F1  64$:   MOVC5   ITEM_SIZE, @ITEM_ADDR, #0, #4, LOCAL_LOCKID      ; 1896
                                 28      AE     004F8                                                                  ;
                                         37  11 004FA  65$:   BRB     69$                                              ; 1877
                     50      2C BE48     7E 004FC  66$:   MOVAQ   @ACP_ATR_LIST[ACP_ATR_PTR], R0                   ; 1903
                02   A0      44 AE4A     F7 00501         CVTLW   ACL_TO_ATR_TAB[ITEM_CODE], 2(R0)                 ;
                     01                  5A  D1 00507         CMPL    ITEM_CODE, #1                                    ; 1904
                                         0F  13 0050A         BEQL    67$                                              ;
                     02                  5A  D1 0050C         CMPL    ITEM_CODE, #2                                    ;
                                         0A  13 0050F         BEQL    67$                                              ;
                     03                  5A  D1 00511         CMPL    ITEM_CODE, #3                                    ; 1905
                                         05  13 00514         BEQL    67$                                              ;
                     06                  5A  D1 00516         CMPL    ITEM_CODE, #6                                    ;
                                         04  12 00519         BNEQ    68$                                              ;
                14   AE                  36  D0 0051B  67$:   MOVL    #54, FUNCTION_CODE                               ; 1906
                             2C BE48     7F 0051F  68$:   PUSHAQ  @ACP_ATR_LIST[ACP_ATR_PTR]                       ; 1907
                     9E          0C      AE  B0 00523         MOVW    ITEM_SIZE, @(SP)+                                ;
                     50      2C BE48     7E 00527         MOVAQ   @ACP_ATR_LIST[ACP_ATR_PTR], R0                   ; 1908
                04   A0          0C      AE  D0 0052C         MOVL    ITEM_ADDR, 4(R0)                                 ;
                                         58  D6 00531         INCL    ACP_ATR_PTR                                      ; 1909
           02        56                  57  F2 00533  69$:   AOBLSS  ITEM_COUNT, J, 70$                               ; 1807
                                         03  11 00537         BRB     71$                                              ;
                                       FEAA  31 00539  70$:   BRW     47$                                              ;
                     50      2C BE48     7E 0053C  71$:   MOVAQ   @ACP_ATR_LIST[ACP_ATR_PTR], R0                   ; 1915
                                 02      A0  B4 00541         CLRW    2(R0)                                            ;
                             2C BE48     7F 00544         PUSHAQ  @ACP_ATR_LIST[ACP_ATR_PTR]                       ; 1916
                                         9E  B4 00548         CLRW    @(SP)+                                           ;
             00C0    CE 00000000'        EF  D0 0054A         MOVL    ACL_CONTEXT, FILE_FIB+48                         ; 1920
                                         7E  D4 00553         CLRL    -(SP)                                            ; 1925
                                 30      AE  DD 00555         PUSHL   ACP_ATR_LIST                                     ;
                                         7E  7C 00558         CLRQ    -(SP)                                            ;
                                         7E  D4 0055A         CLRL    -(SP)                                            ;
                          00E4  CE   9F 0055C        PUSHAB  FILE_FIB_DESC                           :
                                7E   7C 00560        CLRQ    -(SP)                                   :
                            F8  AD   9F 00562        PUSHAB  LOCAL_IOSB                              :
                            38  AE   DD 00565        PUSHL   FUNCTION_CODE                           :
                  7E        48  AE   3C 00568        MOVZWL  IO_CHANNEL, -(SP)                       :
                                7E   D4 0056C        CLRL    -(SP)                                   :
           00000000G  00        0C   FB 0056E        CALLS   #12, SYS$QIOW                           :
                      5B        50   D0 00575        MOVL    R0, STATUS                              :
                      0C        5B   E9 00578        BLBC    STATUS, 72$                             :  1926
                      5B    F8  AD   3C 0057B        MOVZWL  LOCAL_IOSB, STATUS                      :
                      05        5B   E9 0057F        BLBC    STATUS, 72$                             :  1927
                      5B  00C4  CE   D0 00582        MOVL    FILE_FIB+52, STATUS                     :
                            10  AE   B5 00587 72$:   TSTW    LOCAL_CHANNEL                           :  1928
                                0B   12 0058A        BNEQ    73$                                     :
                      7E    20  AE   3C 0058C        MOVZWL  IO_CHANNEL, -(SP)                       :
           00000000G  00        01   FB 00590        CALLS   #1, SYS$DASSGN                          :
                            2C  AE   9F 00597 73$:   PUSHAB  ACP_ATR_LIST                            :  1930
                 18   AE    08  AE   D0 0059A        MOVL    8(SP), 24(SP)                           :
                            18  AE   9F 0059F        PUSHAB  24(SP)                                  :
           00000000G  00        02   FB 005A2        CALLS   #2, LIB$FREE_VM                         :
                      06        5B   E9 005A9        BLBC    STATUS, 74$                             :  1931
                      03        50   E8 005AC        BLBS    STATUS2, 74$                            :
                      5B        50   D0 005AF        MOVL    STATUS2, STATUS                         :
                            28  AE   D5 005B2 74$:   TSTL    LOCAL_LOCKID                            :  1935
                                33   13 005B5        BEQL    77$                                     :
                                7E   7C 005B7        CLRQ    -(SP)                                   :
                                7E   D4 005B9        CLRL    -(SP)                                   :
                            34  AE   DD 005BB        PUSHL   LOCAL_LOCKID                            :
           00000000G  00        04   FB 005BE        CALLS   #4, SYS$DEQ                             :
                                20   11 005C5        BRB     76$                                     :
                 30   AE        03   D0 005C7 75$:   MOVL    #3, CMK_ARG_LIST                        :  1943
                 34   AE        57   D0 005CB        MOVL    ITEM_COUNT, CMK_ARG_LIST+4              :  1944
                 38   AE        59   D0 005CF        MOVL    R9, CMK_ARG_LIST+8                      :  1945
                 3C   AE        5A   9A 005D3        MOVZBL  SHARE, CMK_ARG_LIST+12                  :  1946
                            30  AE   9F 005D7        PUSHAB  CMK_ARG_LIST                            :  1948
                     00000000V  EF   9F 005DA        PUSHAB  ACL_DISPATCH                            :
           00000000G  00        02   FB 005E0        CALLS   #2, SYS$CMKRNL                          :
                      5B        50   D0 005E7 76$:   MOVL    R0, STATUS                              :
                            10  AE   B5 005EA 77$:   TSTW    LOCAL_CHANNEL                           :  1953
                                0B   12 005ED        BNEQ    78$                                     :
                      7E    20  AE   3C 005EF        MOVZWL  IO_CHANNEL, -(SP)                       :
           00000000G  00        01   FB 005F3        CALLS   #1, SYS$DASSGN                          :
                      20    1C  AE   E9 005FA 78$:   BLBC    28(SP), 81$                             :  1957
        1C    BC      04        00   0D 005FE        PROBEW  #0, #4, @CONTEXT                        :  1958
                                16   13 00603        BEQL    80$                                     :
                      08    18  AE   E9 00605        BLBC    24(SP), 79$                             :  1961
                 1C   BC  00C0  CE   D0 00609        MOVL    FILE_FIB+48, @CONTEXT                   :  1962
                                0D   11 0060F        BRB     81$                                     :
                 1C   BC 00000000'  EF   D0 00611 79$:   MOVL    ACL_CONTEXT, @CONTEXT                   :  1963
                                03   11 00619        BRB     81$                                     :  1958
                      5B        0C   D0 0061B 80$:   MOVL    #12, STATUS                             :  1965
                      50        5B   D0 0061E 81$:   MOVL    STATUS, R0                              :  1967
                                     04 00621        RET                                             :  1969

; Routine Size:  1570 bytes,    Routine Base:  $CODE$ + 1318
```

```
%SBTTL  'GET_PARENT_LOCK - get parent for ACL locks'
ROUTINE GET_PARENT_LOCK =

!++
!
! FUNCTIONAL DESCRIPTION:
!
!       This routine takes out a null lock on the system-wide ACL lock
!       parent name. This lock is used as a parent for user mode ACL locks.
!       It must be taken out in kernel mode, since some ACL locks are
!       taken out in kernel mode.
!       numeric value.  If the name does not exist, an error is returned.
!
! CALLING SEQUENCE:
!       GET_PARENT_LOCK ()
!
! INPUT PARAMETERS:
!       none
!
! IMPLICIT INPUTS:
!       none
!
! OUTPUT PARAMETERS:
!       none
!
! IMPLICIT OUTPUTS:
!       PARENT_ID: set to lock ID of parent lock
!
! ROUTINE VALUE:
!       Status of $ENQ call
!
! SIDE EFFECTS:
!       none
!
!--

BEGIN

LOCAL
        STATUS,                         ! system status return
        LOCAL_IOSB      : VECTOR [4, WORD]; ! lock status block


STATUS = $ENQ (LKMODE = LCK$K_NLMODE,
               LKSB   = LOCAL_IOSB,
               RESNAM = .LOCK_PREFIX[0],
               FLAGS  = LCK$M_NOQUEUE OR
                        LCK$M_SYNCSTS OR
                        LCK$M_SYSTEM,
               ACMODE = PSL$C_USER);
IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
IF NOT .STATUS THEN RETURN .STATUS;
PARENT_ID = .(LOCAL_IOSB[2])<0,32>;

1
END;                                    ! End of routine GET_PARENT_LOCK
```

```
                                        0000 00000 GET_PARENT_LOCK:
                                                           .WORD   Save nothing                                          ; 1971
                     5E              08  C2 00002          SUBL2   #8, SP                                                ;
                     7E              03  7D 00005          MOVQ    #3, -(SP)                                             ; 2019
                                     7E  7C 00008          CLRQ    -(SP)                                                 ;
                                     7E  7C 0000A          CLRQ    -(SP)                                                 ;
                       00000000'     EF  DD 0000C          PUSHL   LOCK_PREFIX                                           ;
                                     1C  DD 00012          PUSHL   #28                                                   ;
                              20     AE  9F 00014          PUSHAB  LOCAL_IOSB                                            ;
                                     7E  7C 00017          CLRQ    -(SP)                                                 ;
          00000000G  00              0B  FB 00019          CALLS   #11, SYS$ENQ                                          ;
                     11              50  E9 00020          BLBC    STATUS, 1$                                            ; 2020
                     50              6E  3C 00023          MOVZWL  LOCAL_IOSB, STATUS                                    ;
                     0B              50  E9 00026          BLBC    STATUS, 1$                                            ; 2021
          00000000'  EF       04     AE  D0 00029          MOVL    LOCAL_IOSB+4, PARENT_ID                               ; 2022
                     50              01  D0 00031          MOVL    #1, R0                                                ; 2025
                                     04 00034 1$:          RET                                                           ;

; Routine Size:  53 bytes,    Routine Base:  $CODE$ + 193A
```

```
1 %SBTTL  'SET_ID - TPARSE action routine'
1 ROUTINE SET_ID =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine tries to convert an identifier to its corresponding
1 !       numeric value.  If the name does not exist, an error is returned.
1 !
1 ! CALLING SEQUENCE:
1 !       SET_ID ()
1 !
1 ! INPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT INPUTS:
1 !       ACE_BUFFER: address of the binary ACE storage
1 !       ACE_INDEX: index into the ACE key area
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       ACE_INDEX: index into the ACE key area
1 !
1 ! ROUTINE VALUE:
1 !       SS$_NORMAL if the ID name exists
1 !       SS$_NOSUCHID if it does not
1 !
1 ! SIDE EFFECTS:
1 !       The identifier name is converted to its corresponding value.  That
1 !       value is then placed in the ACE key area.  The index is then updated
1 !       to point to the next available key storage area.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       UIC_POINTER     : REF $BBLOCK;          ! Pointer to UIC entry
2
2 ! Save the identifier, and note the type.
2
2 VECTOR [ACE_BUFFER[ACE$L_KEY], .ACE_INDEX] = .IDENTIFIER;
2 ACE_INDEX = .ACE_INDEX + 1;
2 IF .IDENTIFIER[UIC$V_FORMAT] EQL UIC$K_UIC_FORMAT
2 THEN UIC_COUNT = .UIC_COUNT + 1
2 ELSE ID_COUNT = .ID_COUNT + 1;
2
2 RETURN 1;
2
1 END;                                          ! End of routine SET_ID
```

```
                                        0004 00000 SET_ID: .WORD    Save R2                                          ; 2027
                   52 00000000'     EF   9E 00002          MOVAB    ACE_INDEX, R2
                   50               62   D0 00009          MOVL     ACE_INDEX, R0                                    ; 2070
            FE08 C240          14   A2   D0 0000C          MOVL     IDENTIFIER, ACE_BUFFER+8[R0]
                                    62   D6 00013          INCL     ACE_INDEX                                        ; 2071
              C0   8F          17   A2   93 00015          BITB     IDENTIFIER+3, #192                               ; 2072
                                    05   12 0001A          BNEQ     1$
                               10   A2   D6 0001C          INCL     UIC_COUNT                                        ; 2073
                                    03   11 0001F          BRB      2$
                               20   A2   D6 00021 1$:      INCL     ID_COUNT                                         ; 2074
                   50               01   D0 00024 2$:      MOVL     #1, R0                                           ; 2076
                                         04 00027          RET                                                       ; 2078

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 196F
```

```
1 %SBTTL  'SET_ACCESS_BIT - TPARSE action routine'
1 ROUTINE SET_ACCESS_BIT (ARG1, ARG2, ARG3, SIZE, BUFFER) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine checks the current token to determine whether or not
1 !       it is an access bit name.  If it is, the appropriate bit is set
1 !       in ACE_RIGHTS.  If it is not, an error is returned.
1 !
1 ! CALLING SEQUENCE:
1 !       SET_ACCESS_BIT (ARG1, ARG2, ARG3, ARG4, ARG5)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1-ARG3: TPARSE block arguments not used
1 !       ARG4: size of the current token
1 !       ARG5: address of the current token text
1 !
1 ! IMPLICIT INPUTS:
1 !       none
1 !
1 ! OUTPUT PARAMETERS:
1 !       none
1 !
1 ! IMPLICIT OUTPUTS:
1 !       ACE_RIGHTS
1 !
1 ! ROUTINE VALUE:
1 !       1 if bit name was defined
1 !       0 otherwise
1 !
1 ! SIDE EFFECTS:
1 !       The appropriate bit is set in ACE_RIGHTS.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2       BIT_POSITION,                           ! Bit index
2       BIT_NAME_DESC   : REF $BBLOCK;          ! Bit name descriptor
2
2 ! Note that, initially, no match was found.
2
2 BIT_POSITION = -1;
2
2 ! Now scan the bit name table to see if the specified definition exists.
2
2 INCR J FROM 0 TO 31
2 DO
3     BEGIN
3     IF .BIT_NAME_TABLE NEQA 0
3     THEN
4       BEGIN
4       IF PROBER (%REF (0), %REF (DSC$C_S_BLN), BIT_NAME_TABLE[.J, 0, 0, 0, 0])
4       THEN BIT_NAME_DESC = BIT_NAME_TABLE[.J, 0, 0, 0, 0]
```

```
; 2144      2136  4             ELSE RETURN SS$_ACCVIO;
; 2145      2137  4             IF NOT EXE$PROBER (0, .BIT_NAME_DESC[DSC$W_LENGTH],
; 2146      2138  4                             .BIT_NAME_DESC[DSC$A_POINTER])
; 2147      2139  4             THEN RETURN SS$_ACCVIO;
; 2148      2140  4             END
; 2149      2141  3         ELSE BIT_NAME_DESC = .DEFAULT_BITS[.J];
; 2150      2142  3         IF CH$EQL (.SIZE, .BUFFER,
; 2151      2143  3                 MINU (.SIZE, .BIT_NAME_DESC[DSC$W_LENGTH]), .BIT_NAME_DESC[DSC$A_POINTER], 0)
; 2152      2144  3         THEN
; 2153      2145  4             BEGIN
; 2154      2146  4             IF .BIT_POSITION GEQ 0 THEN RETURN 0;   ! Ambiguous, error.
; 2155      2147  4             BIT_POSITION = .J;
; 2156      2148  3             END;
; 2157      2149  2         END;
; 2158      2150  2
; 2159      2151  2     IF .BIT_POSITION LSS 0 THEN RETURN 0;              ! Specified name not found
; 2160      2152  2     ACE_RIGHTS<.BIT_POSITION,1> = 1;                   ! Note desired access.
; 2161      2153  2
; 2162      2154  2     RETURN 1;
; 2163      2155  2
; 2164      2156  1     END;                                               ! End of routine SET_ACCESS_BIT



                                               007C 00000 SET_ACCESS_BIT:
                                                                .WORD   Save R2,R3,R4,R5,R6                      ; 2080
                     56             01    CE 00002              MNEGL   #1, BIT_POSITION                         ; 2124
                                    54    D4 00005              CLRL    J                                        ; 2142
                     50 00000000'   EF    D0 00007 1$:          MOVL    BIT_NAME_TABLE, R0                       ; 2131
                                    23    13 0000E              BEQL    3$                                       ;
                                  6044    7F 00010              PUSHAQ  (R0)[J]                                  ; 2134
            9E       08             00    0C 00013              PROBER  #0, #8, @(SP)+                           ;
                                    16    13 00017              BEQL    2$                                       ;
                     55           6044    7E 00019              MOVAQ   (R0)[J], BIT_NAME_DESC                   ; 2135
                     50        04   A5    D0 0001D              MOVL    4(BIT_NAME_DESC), R0                     ; 2137
                     51             65    3C 00021              MOVZWL  (BIT_NAME_DESC), R1                      ;
                                    53    D4 00024              CLRL    R3                                       ;
                        00000000G   00    16 00026              JSB     EXE$PROBER                               ;
                     0C             50    E8 0002C              BLBS    R0, 4$                                   ;
                     50             0C    D0 0002F 2$:          MOVL    #12, R0                                  ; 2139
                                          04 00032              RET                                              ;
                     55 00000000'EF44     D0 00033 3$:          MOVL    DEFAULT_BITS[J], BIT_NAME_DESC           ; 2141
                     50        10   AC    D0 0003B 4$:          MOVL    SIZE, R0                                 ; 2143
   50       65       10             00    ED 0003F              CMPZV   #0, #16, (BIT_NAME_DESC), R0             ;
                                    03    1E 00044              BGEQU   5$                                       ;
                     50             65    3C 00046              MOVZWL  (BIT_NAME_DESC), R0                      ;
   50       00   14  BC        10   AC    2D 00049 5$:          CMPC5   SIZE, @BUFFER, #0, R0, @4(BIT_NAME_DESC) ; 2142
                               04   B5       00050                                                               ;
                                    07    12 00052              BNEQ    6$                                       ;
                                    56    D5 00054              TSTL    BIT_POSITION                             ; 2146
                                    17    18 00056              BGEQ    8$                                       ;
                     56             54    D0 00058              MOVL    J, BIT_POSITION                          ; 2147
            A8       54             1F    F3 0005B 6$:          AOBLEQ  #31, J, 1$                               ; 2128
                                    56    D5 0005F              TSTL    BIT_POSITION                             ; 2151
                                    0C    19 00061              BLSS    8$                                       ;
```

```
           00 00000000'  EF       56  E2 00063        BBSS    BIT_POSITION, ACE_RIGHTS, 7$                    ; 2152
                         50       01  D0 0006B 7$:     MOVL    #1, R0                                          ; 2154
                                      04 0006E         RET                                                     ;
                                  50  D4 0006F 8$:     CLRL    R0                                              ; 2156
                                      04 00071         RET                                                     ;

; Routine Size:  114 bytes,    Routine Base:  $CODE$ + 1997
```

```
1 %SBTTL  'GET_UCB_ACL - get UCB ACL queue head address'
1 ROUTINE GET_UCB_ACL (CHANNEL) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !        This routine locates the ACL queue head for a device, given a
1 !        channel number.
1 !
1 ! CALLING SEQUENCE:
1 !        GET_UCB_ACL (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !        ARG1: channel assigned to the device
1 !
1 ! IMPLICIT INPUTS:
1 !        none
1 !
1 ! OUTPUT PARAMETERS:
1 !        none
1 !
1 ! IMPLICIT OUTPUTS:
1 !        none
1 !
1 ! ROUTINE VALUE:
1 !        SS$_NORMAL if ACLs are allowed
1 !        SS$_NOACLSUPPORT if ACLs are not allowed
1 !
1 ! SIDE EFFECTS:
1 !        none
1 !
1 !--
1
2 BEGIN
2
2 MAP
2         CHANNEL          : WORD;
2
2 LOCAL
2         STATUS,                                  ! Local routine return status
2         DEVICE_UCB       : REF $BBLOCK,          ! Device UCB address
2         DEVICE_ORB       : REF $BBLOCK,          ! Device ORB address
2         CHANNEL_BLOCK    : REF $BBLOCK;          ! Device CCB address
2
2 STATUS = IOC$VERIFYCHAN (.CHANNEL; CHANNEL_BLOCK);
2 IF NOT .STATUS THEN RETURN .STATUS;
2 DEVICE_UCB = .CHANNEL_BLOCK[CCB$L_UCB];
2 DEVICE_ORB = .DEVICE_UCB[UCB$L_ORB];
2
2 ! If no ACLs are allowed, return an error.
2
2 IF .DEVICE_ORB[ORB$V_NOACL] THEN RETURN SS$_NOACLSUPPORT;
2
2 ! If the device is unowned, and the protection is all access to everybody,
2 !  and there is no ACL present, require SYSPRV to change the ACL.
2
```

```
; 2223      2214  2 IF .DEVICE_ORB[ORB$L_OWNER] EQL 0
; 2224      2215  3 AND (IF .DEVICE_ORB[ORB$V_ACL_QUEUE]
; 2225      2216  3      THEN .DEVICE_ORB[ORB$L_ACLFL] EQLA DEVICE_ORB[ORB$L_ACLFL]
; 2226      2217  3      ELSE 1)
; 2227      2218  2 AND NOT .$BBLOCK [CTL$GL_PCB[PCB$Q_PRIV], PRV$V_SYSPRV]
; 2228      2219  2 THEN RETURN SS$_NOPRIV;
; 2229      2220  2
; 2230      2221  2 ! If the ACL queue head is uninitialized, initialize it now.
; 2231      2222  2
; 2232      2223  2 IF NOT .DEVICE_ORB[ORB$V_ACL_QUEUE] THEN ACL_INIT_QUEUE (.DEVICE_ORB);
; 2233      2224  2
; 2234      2225  2 ! Set up the address of the ACL queue head.
; 2235      2226  2
; 2236      2227  2 ACL_QUEUE_HEAD = $BBLOCK [.DEVICE_UCB[UCB$L_ORB], ORB$L_ACLFL];
; 2237      2228  2
; 2238      2229  2 RETURN SS$_NORMAL;
; 2239      2230  2
; 2240      2231  1 END;                                                  ! End of routine GET_UCB_ACL



                                                   000C 00000 GET_UCB_ACL:
                                                                     .WORD   Save R2,R3                                   ; 2158
                           50          04     AC    3C 00002         MOVZWL  CHANNEL, R0                                  ; 2202
                              00000000G       00    16 00006         JSB     IOC$VERIFYCHAN
                           50                 50    E9 0000C         BLBC    STATUS, 5$                                   ; 2203
                           52                 61    D0 0000F         MOVL    (CHANNEL_BLOCK), DEVICE_UCB                  ; 2204
                           50          1C     A2    D0 00012         MOVL    28(DEVICE_UCB), DEVICE_ORB                   ; 2205
             06       0B   A0                 03    E1 00016         BBC     #3, 11(DEVICE_ORB), 1$                       ; 2209
                           50        22BC     8F    3C 0001B         MOVZWL  #8892, R0                                    ;
                                                    04 00020         RET                                                  ;
                                              60    D5 00021 1$:     TSTL    (DEVICE_ORB)                                 ; 2214
                                              20    12 00023         BNEQ    3$                                           ;
             0A       0B   A0                 01    E1 00025         BBC     #1, 11(DEVICE_ORB), 2$                       ; 2215
                           51          28     A0    9E 0002A         MOVAB   40(DEVICE_ORB), R1                           ; 2216
                           51          28     A0    D1 0002E         CMPL    40(DEVICE_ORB), R1                           ;
                                              11    12 00032         BNEQ    3$                                           ;
                           51   00000000G     00    D0 00034 2$:     MOVL    CTL$GL_PCB, R1                               ; 2218
             04     0087   C1                 04    E0 0003B         BBS     #4, 135(R1), 3$                              ;
                           50                 24    D0 00041         MOVL    #36, R0                                      ; 2219
                                              04 00044               RET                                                  ;
             09       0B   A0                 01    E0 00045 3$:     BBS     #1, 11(DEVICE_ORB), 4$                       ; 2223
                                              50    DD 0004A         PUSHL   DEVICE_ORB                                   ;
                    00000000G      00         01    FB 0004C         CALLS   #1, ACL_INIT_QUEUE                           ;
   00000000'   EF          1C      A2         28    C1 00053 4$:     ADDL3   #40, 28(DEVICE_UCB), ACL_QUEUE_HEAD          ; 2227
                           50                 01    D0 0005C         MOVL    #1, R0                                       ; 2229
                                                    04 0005F 5$:     RET                                                  ; 2231

; Routine Size:  96 bytes,    Routine Base:  $CODE$ + 1A09
; 2242          2232  1 %SBTTL  'GET_JBC_ACL - get Job Controller queue ACL queue head address'
; 2243          2233  1 ROUTINE GET_JBC_ACL (QUEUE_NAME) =
; 2244          2234  1
; 2245          2235  1 !++
; 2246          2236  1 !
; 2247          2237  1 ! FUNCTIONAL DESCRIPTION:
; 2248          2238  1 !
; 2249          2239  1 !       This routine locates the ACL queue head for a Job Controller
; 2250          2240  1 !       queue.
; 2251          2241  1 !
; 2252          2242  1 ! CALLING SEQUENCE:
; 2253          2243  1 !       GET_JBC_ACL (ARG1)
; 2254          2244  1 !
; 2255          2245  1 ! INPUT PARAMETERS:
; 2256          2246  1 !       ARG1: address of the queue name descriptor
; 2257          2247  1 !
; 2258          2248  1 ! IMPLICIT INPUTS:
; 2259          2249  1 !       none
; 2260          2250  1 !
; 2261          2251  1 ! OUTPUT PARAMETERS:
; 2262          2252  1 !       none
; 2263          2253  1 !
; 2264          2254  1 ! IMPLICIT OUTPUTS:
; 2265          2255  1 !       none
; 2266          2256  1 !
; 2267          2257  1 ! ROUTINE VALUE:
; 2268          2258  1 !       address of the ACL queue head or 0 if an error has occurred
; 2269          2259  1 !
; 2270          2260  1 ! SIDE EFFECTS:
; 2271          2261  1 !       none
; 2272          2262  1 !
; 2273          2263  1 !--
; 2274          2264  1
; 2275          2265  2 BEGIN
; 2276          2266  2
; 2277          2267  2 MAP
; 2278          2268  2       QUEUE_NAME      : REF $BBLOCK;
; 2279          2269  2
; 2280          2270  2 RETURN SS$_BADPARAM;
; 2281          2271  2
; 2282          2272  1 END;                                        ! End of routine GET_JBC_ACL



                                       0000 00000 GET_JBC_ACL:
                                                          .WORD   Save nothing                                  ; 2233
                          50          14  D0 00002        MOVL    #20, R0                                       ; 2270
                                          04 00005        RET                                                   ; 2272

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 1A69
; 2284          2273  1 %SBTTL  'GET_CEB_ACL - get common event block ACL queue head address'
; 2285          2274  1 ROUTINE GET_CEB_ACL (CLUSTER_NAME) =
; 2286          2275  1
; 2287          2276  1 !++
; 2288          2277  1 !
; 2289          2278  1 ! FUNCTIONAL DESCRIPTION:
; 2290          2279  1 !
; 2291          2280  1 !       This routine locates the ACL queue head for a common event block.
; 2292          2281  1 !
; 2293          2282  1 ! CALLING SEQUENCE:
; 2294          2283  1 !       GET_CEB_ACL (ARG1)
; 2295          2284  1 !
; 2296          2285  1 ! INPUT PARAMETERS:
; 2297          2286  1 !       ARG1: address of the cluster name descriptor
; 2298          2287  1 !
; 2299          2288  1 ! IMPLICIT INPUTS:
; 2300          2289  1 !       none
; 2301          2290  1 !
; 2302          2291  1 ! OUTPUT PARAMETERS:
; 2303          2292  1 !       none
; 2304          2293  1 !
; 2305          2294  1 ! IMPLICIT OUTPUTS:
; 2306          2295  1 !       none
; 2307          2296  1 !
; 2308          2297  1 ! ROUTINE VALUE:
; 2309          2298  1 !       address of the ACL queue head or 0 if an error has occurred
; 2310          2299  1 !
; 2311          2300  1 ! SIDE EFFECTS:
; 2312          2301  1 !       none
; 2313          2302  1 !
; 2314          2303  1 !--
; 2315          2304  1
; 2316          2305  2 BEGIN
; 2317          2306  2
; 2318          2307  2 MAP
; 2319          2308  2       CLUSTER_NAME    : REF $BBLOCK;
; 2320          2309  2
; 2321          2310  2 RETURN SS$_BADPARAM;
; 2322          2311  2
; 2323          2312  1 END;                                          ! End of routine GET_CEB_ACL



                                    0000 00000 GET_CEB_ACL:
                                                      .WORD    Save nothing                                      ; 2274
                         50             14  D0 00002  MOVL     #20, R0                                           ; 2310
                                            04 00005  RET                                                        ; 2312

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 1A6F
; 2325      2313  1 %SBTTL  'GET_LNT_ACL - get logical name table ACL queue head address'
; 2326      2314  1 ROUTINE GET_LNT_ACL (TABLE_NAME) =
; 2327      2315  1
; 2328      2316  1 !++
; 2329      2317  1 !
; 2330      2318  1 ! FUNCTIONAL DESCRIPTION:
; 2331      2319  1 !
; 2332      2320  1 !       This routine locates the ACL queue head for aLogical name
; 2333      2321  1 !       table.
; 2334      2322  1 !
; 2335      2323  1 ! CALLING SEQUENCE:
; 2336      2324  1 !       GET_LNT_ACL (ARG1)
; 2337      2325  1 !
; 2338      2326  1 ! INPUT PARAMETERS:
; 2339      2327  1 !       ARG1: address of the table name descriptor
; 2340      2328  1 !
; 2341      2329  1 ! IMPLICIT INPUTS:
; 2342      2330  1 !       none
; 2343      2331  1 !
; 2344      2332  1 ! OUTPUT PARAMETERS:
; 2345      2333  1 !       none
; 2346      2334  1 !
; 2347      2335  1 ! IMPLICIT OUTPUTS:
; 2348      2336  1 !       none
; 2349      2337  1 !
; 2350      2338  1 ! ROUTINE VALUE:
; 2351      2339  1 !       address of the ACL queue head or 0 if an error has occurred
; 2352      2340  1 !
; 2353      2341  1 ! SIDE EFFECTS:
; 2354      2342  1 !       none
; 2355      2343  1 !
; 2356      2344  1 !--
; 2357      2345  1
; 2358      2346  2 BEGIN
; 2359      2347  2
; 2360      2348  2 RETURN SS$_BADPARAM;
; 2361      2349  2
; 2362      2350  1 END;                                              ! End of routine GET_LNT_ACL


                                      0000 00000 GET_LNT_ACL:
                                                        .WORD   Save nothing                                    ; 2314
                    50                   14  D0 00002   MOVL    #20, R0                                         ; 2348
                                             04 00005   RET                                                     ; 2350

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 1A75
; 2364       2351  1 %SBTTL  'GET_PCB_ACL - get process ACL queue head address'
; 2365       2352  1 ROUTINE GET_PCB_ACL (PROCESS_NAME) =
; 2366       2353  1
; 2367       2354  1 !++
; 2368       2355  1 !
; 2369       2356  1 ! FUNCTIONAL DESCRIPTION:
; 2370       2357  1 !
; 2371       2358  1 !       This routine locates the ACL queue head for a process.
; 2372       2359  1 !
; 2373       2360  1 ! CALLING SEQUENCE:
; 2374       2361  1 !       GET_PCB_ACL (ARG1)
; 2375       2362  1 !
; 2376       2363  1 ! INPUT PARAMETERS:
; 2377       2364  1 !       ARG1: address of the process name descriptor
; 2378       2365  1 !
; 2379       2366  1 ! IMPLICIT INPUTS:
; 2380       2367  1 !       none
; 2381       2368  1 !
; 2382       2369  1 ! OUTPUT PARAMETERS:
; 2383       2370  1 !       none
; 2384       2371  1 !
; 2385       2372  1 ! IMPLICIT OUTPUTS:
; 2386       2373  1 !       none
; 2387       2374  1 !
; 2388       2375  1 ! ROUTINE VALUE:
; 2389       2376  1 !       address of the ACL queue head or 0 if an error has occurred
; 2390       2377  1 !
; 2391       2378  1 ! SIDE EFFECTS:
; 2392       2379  1 !       none
; 2393       2380  1 !
; 2394       2381  1 !--
; 2395       2382  1
; 2396       2383  2 BEGIN
; 2397       2384  2
; 2398       2385  2 RETURN SS$_BADPARAM;
; 2399       2386  2
; 2400       2387  1 END;                                           ! End of routine GET_PCB_ACL



                                    0000 00000 GET_PCB_ACL:
                                                      .WORD    Save nothing                                    ; 2352
                    50            14  D0 00002        MOVL     #20, R0                                         ; 2385
                                      04 00005        RET                                                      ; 2387

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 1A7B
; 2402          2388  1 %SBTTL  'GET_GBL_ACL - get global section ACL queue head address'
; 2403          2389  1 ROUTINE GET_GBL_ACL (SECTION_NAME) =
; 2404          2390  1
; 2405          2391  1 !++
; 2406          2392  1 !
; 2407          2393  1 ! FUNCTIONAL DESCRIPTION:
; 2408          2394  1 !
; 2409          2395  1 !       This routine locates the ACL queue head for a global section, given
; 2410          2396  1 !       a section name.
; 2411          2397  1 !
; 2412          2398  1 ! CALLING SEQUENCE:
; 2413          2399  1 !       GET_GBL_ACL (ARG1)
; 2414          2400  1 !
; 2415          2401  1 ! INPUT PARAMETERS:
; 2416          2402  1 !       ARG1: address of the section name descriptor
; 2417          2403  1 !
; 2418          2404  1 ! IMPLICIT INPUTS:
; 2419          2405  1 !       none
; 2420          2406  1 !
; 2421          2407  1 ! OUTPUT PARAMETERS:
; 2422          2408  1 !       none
; 2423          2409  1 !
; 2424          2410  1 ! IMPLICIT OUTPUTS:
; 2425          2411  1 !       none
; 2426          2412  1 !
; 2427          2413  1 ! ROUTINE VALUE:
; 2428          2414  1 !       address of the ACL queue head or 0 if an error has occurred
; 2429          2415  1 !
; 2430          2416  1 ! SIDE EFFECTS:
; 2431          2417  1 !       none
; 2432          2418  1 !
; 2433          2419  1 !--
; 2434          2420  1
; 2435          2421  2 BEGIN
; 2436          2422  2
; 2437          2423  2 RETURN SS$_BADPARAM;
; 2438          2424  2
; 2439          2425  1 END;                                        ! End of routine GET_GBL_ACL



                                        0000 00000 GET_GBL_ACL:
                                                         .WORD   Save nothing                                    ; 2389
                             50            14  D0 00002  MOVL    #20, R0                                         ; 2423
                                               04 00005  RET                                                     ; 2425

; Routine Size:  6 bytes,    Routine Base:  $CODE$ + 1A81
```

```
1 %SBTTL  'ACL_DISPATCH - main ACL function dispatcher'
1 ROUTINE ACL_DISPATCH (ITEM_COUNT, ITEM_LIST, SHARE) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called to perform the appropriate ACL operations.
1 !       The code is checked for validity and, when necessary, the buffer
1 !       is probed for the desired access.
1 !
1 ! CALLING SEQUENCE:
1 !       ACL_DISPATCH (ARG1, ARG2, ARG3)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: count of items to process
1 !       ARG2: address of the item list
1 !       ARG3: 1 if the operation is only reading the ACL
1 !             0 if the ACL is being modified
1 !
1 ! IMPLICIT INPUTS:
1 !       ACL_CONTEXT: previous ACL context
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       ACL_CONTEXT: new ACL context
1 !
1 ! ROUTINE VALUE:
1 !       1
1 !
1 ! SIDE EFFECTS:
1 !       The appropriate action routine is called.  Possible ACL modification
1 !       may result.
1 !
1 !--
1
2 BEGIN
2
2 MAP
2       ITEM_LIST       : REF BLOCKVECTOR [, ITM$S_ITEM, BYTE];
2
2 ! Cells defined to tie off references made in the module ALLOCB obtained from
2 ! the XQP.
2
2 GLOBAL  LITERAL
2       CONTEXT_SAVE    = 0,
2       CURRENT_WINDOW  = 0,
2       IO_PACKET       = 0;
2
2 LOCAL
2       ACL_STATUS,                             ! Status returned by ACL operation
2       STATUS,                                 ! Routine return status
2       FUNCTION_CODE,                          ! Operation to perform
2       SIZE,                                   ! Size of user buffer
2       BUFFER          : REF $BBLOCK,          ! Address of user buffer
```

```
2               LOCAL_IOSB      : VECTOR [4, WORD],        ! Lock status block
2               LOCAL_LOCKID;                              ! Local copy of the lock-id
2
2       ! Initialize local storage.
2
2       CH$FILL (0, 4*2, LOCAL_IOSB);
2       LOCAL_IOSB[0] = SS$_NORMAL;                        ! Assume success
2       ACL_STATUS = STATUS = SS$_NORMAL;                  ! Here also
2
2       ! Take out the mutex on the specified ACL.
2
2       IF .SHARE
2       THEN SCH$LOCKR (.ACL_QUEUE_HEAD - $BYTEOFFSET (ORB$L_ACLFL) + $BYTEOFFSET (ORB$L_ACL_MUTEX), .CTL$GL_PCB)
2       ELSE SCH$LOCKW (.ACL_QUEUE_HEAD - $BYTEOFFSET (ORB$L_ACLFL) + $BYTEOFFSET (ORB$L_ACL_MUTEX), .CTL$GL_PCB);
2
2       ! Loop over the item list, processing each item.
2
2       INCR J FROM 0 TO .ITEM_COUNT-1
2       DO
3           BEGIN
3
3           FUNCTION_CODE = .ITEM_LIST[.J, ITM$W_ITMCOD];
3           SIZE = .ITEM_LIST[.J, ITM$W_BUFSIZ];
3           BUFFER = .ITEM_LIST[.J, ITM$L_BUFADR];
3
3       ! Dispatch on the function code.
3
3           CASE .FUNCTION_CODE FROM MIN_ACL_ATR TO MAX_ACL_ATR OF
3           SET
3
3               [ACL$C_ADDACLENT]:
4                   BEGIN
4                   IF .SHARE
4                   THEN STATUS = SS$_BADPARAM
4                   ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
4                   THEN STATUS = SS$_ACCVIO
4                   ELSE IF .ACL_STATUS
4                   THEN ACL_STATUS = ACL_ADDENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
3                   END;
3
3               [ACL$C_DELACLENT]:
4                   BEGIN
4                   IF .SHARE
4                   THEN STATUS = SS$_BADPARAM
4                   ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
4                   THEN STATUS = SS$_ACCVIO
4                   ELSE IF .ACL_STATUS
4                   THEN ACL_STATUS = ACL_DELENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
3                   END;
3
3               [ACL$C_MODACLENT]:
4                   BEGIN
4                   IF .SHARE
4                   THEN STATUS = SS$_BADPARAM
4                   ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
4                   THEN STATUS = SS$_ACCVIO
4                   ELSE IF .ACL_STATUS
```

```
4                       THEN ACL_STATUS = ACL_MODENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
3                       END;
3
3               [ACL$C_FNDACLENT]:
4                       BEGIN
4                       IF NOT EXE$PROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
4                       THEN STATUS = SS$_ACCVIO
4                       ELSE ACL_STATUS = ACL_FINDENTRY (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER, 0);
3                       END;
3
3               [ACL$C_FNDACETYP]:
4                       BEGIN
4                       IF NOT EXE$PROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
4                       THEN STATUS = SS$_ACCVIO
4                       ELSE ACL_STATUS = ACL_FINDTYPE (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER, 0);
3                       END;
3
3               [ACL$C_DELETEACL]:
4                       BEGIN
4                       IF .SHARE
4                       THEN STATUS = SS$_BADPARAM
4                       ELSE IF .ACL_STATUS
4                       THEN ACL_STATUS = ACL_DELETEACL (.ACL_QUEUE_HEAD, ACL_CONTEXT);
3                       END;
3
3               [ACL$C_READACL]:
4                       BEGIN
4                       IF NOT EXE$PROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
4                       THEN STATUS = SS$_ACCVIO
4                       ELSE ACL_STATUS = ACL_READACL (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
3                       END;
3
3               [ACL$C_ACLLENGTH]:
4                       BEGIN
4                       IF NOT EXE$PROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
4                       THEN STATUS = SS$_ACCVIO
4                       ELSE ACL_STATUS = ACL_ACLLENGTH (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
3                       END;
3
3               [ACL$C_READACE]:
4                       BEGIN
4                       IF NOT EXE$PROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
4                       THEN STATUS = SS$_ACCVIO
4                       ELSE ACL_STATUS = ACL_READACE (.ACL_QUEUE_HEAD, ACL_CONTEXT, .SIZE, .BUFFER);
3                       END;
3
3
3               [ACL$C_RLOCK_ACL,
3                ACL$C_WLOCK_ACL]:
4                       BEGIN
4                       IF .SIZE LSSU 4
4                       THEN STATUS = SS$_BADPARAM
4                       ELSE IF NOT EXE$PROBEW (.CALL_ACMODE, .SIZE, .BUFFER)
4                       THEN STATUS = SS$_ACCVIO
4                       ELSE
5                           BEGIN
5  P                        STATUS = $ENQ (LKMODE = (IF .FUNCTION_CODE EQL ACL$C_RLOCK_ACL
                                        THEN LCK$K_CRMODE ELSE LCK$K_PWMODE),
                              LKSB = LOCAL_IOSB,
                              RESNAM = LOCK_RESNAM,
                              PARID = (IF .CALL_ACMODE EQL PSL$C_USER
                                       THEN .PARENT_ID
                                       ELSE 0),
                              FLAGS = LCK$M_NOQUEUE OR
                                      LCK$M_SYNCSTS OR
                                      LCK$M_SYSTEM,
                              ACMODE = PSL$C_USER);
                IF .STATUS THEN STATUS = .LOCAL_IOSB[0];
                CH$COPY (4, LOCAL_IOSB[2],
                        0,
                        .SIZE, .BUFFER);                      ! Copy lock-id
                END;
            END;


        [ACL$C_UNLOCK_ACL]:
            BEGIN
            IF .SIZE LSSU 4
            THEN STATUS = SS$_BADPARAM
            ELSE IF NOT EXE$PROBER (.CALL_ACMODE, .SIZE, .BUFFER)
            THEN STATUS = SS$_ACCVIO
            ELSE
                BEGIN
                CH$COPY (.SIZE, .BUFFER, 0, 4, LOCAL_LOCKID);
                STATUS = $DEQ (LKID = .LOCAL_LOCKID);
                END;
            END;

        [INRANGE, OUTRANGE]:
            BEGIN
            STATUS = SS$_BADPARAM;
            END;

    TES;

    IF NOT .STATUS THEN EXITLOOP;
    END;

! If an error occurred because of an access violation or an access conflict
! (trying to modify the ACL when only holding a read lock), return it.  Otherwise
! return any error that may have occurred during the ACL processing.

IF .STATUS THEN STATUS = .ACL_STATUS;

! Release the ACL mutex.

SCH$UNLOCK (.ACL_QUEUE_HEAD - $BYTEOFFSET (ORB$L_ACLFL) + $BYTEOFFSET (ORB$L_ACL_MUTEX), .CTL$GL_PCB);

RETURN .STATUS;

END;                                          ! End of routine ACL_DISPATCH
```

```
                                                                  K  4
SYSACLSRV                                                         16-Sep-1984 01:51:51    VAX-11 Bliss-32 V4.0-742              Page 105
V04-000         ACL_DISPATCH - main ACL function dispatcher       14-Sep-1984 12:40:53    [LOADSS.SRC]SYSACLSRV.B32;1               (16)

                                                                  CONTEXT_SAVE==         0
                                                                  CURRENT_WINDOW==       0
                                                                  IO_PACKET==            0

                                                      0FFC 00000 ACL_DISPATCH:
                                                                         .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 2427
                                      5E          10  C2 00002           SUBL2    #16, SP                                         ;
        08          00                6E          00  2C 00005           MOVC5    #0, (SP), #0, #8, LOCAL_IOSB                    ; 2488
                                            08    AE     0000A                                                                    ;
                              08      AE          01  B0 0000C           MOVW     #1, LOCAL_IOSB                                  ; 2489
                                      59          01  D0 00010           MOVL     #1, STATUS                                      ; 2490
                                      5B          01  D0 00013           MOVL     #1, ACL_STATUS                                  ;
                                      54 00000000G  00  D0 00016         MOVL     CTL$GL_PCB, R4                                  ; 2495
                    50 00000000'      EF          24  C3 0001D           SUBL3    #36, ACL_QUEUE_HEAD, R0                         ;
                                      6E    0C    AC  D0 00025           MOVL     SHARE, (SP)                                     ; 2494
                                      08          6E  E9 00029           BLBC     (SP), 1$                                        ;
                            00000000G 00          16  0002C              JSB      SCH$LOCKR                                       ; 2495
                                      06          11  00032              BRB      2$                                              ;
                            00000000G 00          16  00034  1$:         JSB      SCH$LOCKW                                       ; 2496
                                      56          01  CE 0003A  2$:      MNEGL    #1, J                                           ; 2504
                                            0291  31  0003D              BRW      46$                                             ;
                    51                56          0C  C5 00040  3$:      MULL3    #12, J, R1                                      ;
                    50                51    08    AC  C1 00044           ADDL3    ITEM_LIST, R1, R0                               ;
                                      5A    02    A0  3C 00049           MOVZWL   2(R0), FUNCTION_CODE                            ;
                    50                51    08    AC  C1 0004D           ADDL3    ITEM_LIST, R1, R0                               ; 2505
                                      57          60  3C 00052           MOVZWL   (R0), SIZE                                      ;
                    50                51    08    AC  C1 00055           ADDL3    ITEM_LIST, R1, R0                               ; 2506
                                      58    04    A0  D0 0005A           MOVL     4(R0), BUFFER                                   ;
                    0B                01          5A  CF 0005E           CASEL    FUNCTION_CODE, #1, #11                          ; 2510
        00BA        0083              004F        001B     00062  4$:    .WORD    6$-4$,-                                         ;
        0169        013B              011A        00EA     0006A                  7$-4$,-                                         ;
        022F        01C8              01C8        0197     00072                  8$-4$,-                                         ;
                                                                                  13$-4$,-                                        ;
                                                                                  16$-4$,-                                        ;
                                                                                  18$-4$,-                                        ;
                                                                                  23$-4$,-                                        ;
                                                                                  26$-4$,-                                        ;
                                                                                  29$-4$,-                                        ;
                                                                                  32$-4$,-                                        ;
                                                                                  32$-4$,-                                        ;
                                                                                  39$-4$                                          ;
                                            0219  31  0007A  5$:         BRW      40$                                             ; 2630
                                      FA          6E  E8 0007D  6$:      BLBS     (SP), 5$                                        ; 2515
                                      50          58  D0 00080           MOVL     BUFFER, R0                                      ; 2517
                                      51          57  D0 00083           MOVL     SIZE, R1                                        ;
                                      53 00000000'  EF  D0 00086         MOVL     CALL_ACMODE, R3                                 ;
                                         00000000G  00  16 0008D         JSB      EXE$PROBER                                      ;
                                      65          50  E9 00093           BLBC     R0, 9$                                          ;
                                      65          5B  E9 00096           BLBC     ACL_STATUS, 10$                                 ; 2519
                                      7E          57  7D 00099           MOVQ     SIZE, -(SP)                                     ; 2520
                                         00000000'  EF  9F 0009C         PUSHAB   ACL_CONTEXT                                     ;
                                         00000000'  EF  DD 000A2         PUSHL    ACL_QUEUE_HEAD                                  ;
                        00000000G     00          04  FB 000A8           CALLS    #4, ACL_ADDENTRY                                ;
                                      69          11  000AF              BRB      12$                                             ;
                                      C6          6E  E8 000B1  7$:      BLBS     (SP), 5$                                        ; 2525
                                      50          58  D0 000B4           MOVL     BUFFER, R0                                      ; 2527
```

```
                      51          57  D0 000B7        MOVL    SIZE, R1                                          ;
                      53  00000000'  EF  D0 000BA        MOVL    CALL_ACMODE, R3                                   ;
                          00000000G  00  16 000C1        JSB     EXE$PROBER                                        ;
                      65          50  E9 000C7        BLBC    R0, 14$                                           ;
                      31          5B  E9 000CA        BLBC    ACL_STATUS, 10$                                   ; 2529
                      7E          57  7D 000CD        MOVQ    SIZE, -(SP)                                       ; 2530
                          00000000'  EF  9F 000D0        PUSHAB  ACL_CONTEXT                                       ;
                          00000000'  EF  DD 000D6        PUSHL   ACL_QUEUE_HEAD                                    ;
              00000000G   00          04  FB 000DC        CALLS   #4, ACL_DELENTRY                                  ;
                                      65  11 000E3        BRB     15$                                               ;
                      92          6E  E8 000E5  8$:   BLBS    (SP), 5$                                          ; 2535
                      50          58  D0 000E8        MOVL    BUFFER, R0                                        ; 2537
                      51          57  D0 000EB        MOVL    SIZE, R1                                          ;
                      53  00000000'  EF  D0 000EE        MOVL    CALL_ACMODE, R3                                   ;
                          00000000G  00  16 000F5        JSB     EXE$PROBER                                        ;
                      61          50  E9 000FB  9$:   BLBC    R0, 17$                                           ;
                      03          5B  E8 000FE  10$:  BLBS    ACL_STATUS, 11$                                   ; 2539
                                0081  31 00101        BRW     20$                                               ;
                      7E          57  7D 00104  11$:  MOVQ    SIZE, -(SP)                                       ; 2540
                          00000000'  EF  9F 00107        PUSHAB  ACL_CONTEXT                                       ;
                          00000000'  EF  DD 0010D        PUSHL   ACL_QUEUE_HEAD                                    ;
              00000000G   00          04  FB 00113        CALLS   #4, ACL_MODENTRY                                  ;
                                      7F  11 0011A  12$:  BRB     22$                                               ;
                      50          58  D0 0011C  13$:  MOVL    BUFFER, R0                                        ; 2545
                      51          57  D0 0011F        MOVL    SIZE, R1                                          ;
                      53  00000000'  EF  D0 00122        MOVL    CALL_ACMODE, R3                                   ;
                          00000000G  00  16 00129        JSB     EXE$PROBEW                                        ;
                      7E          50  E9 0012F  14$:  BLBC    R0, 24$                                           ;
                                  7E  D4 00132        CLRL    -(SP)                                             ; 2547
                      7E          57  7D 00134        MOVQ    SIZE, -(SP)                                       ;
                          00000000'  EF  9F 00137        PUSHAB  ACL_CONTEXT                                       ;
                          00000000'  EF  DD 0013D        PUSHL   ACL_QUEUE_HEAD                                    ;
              00000000G   00          05  FB 00143        CALLS   #5, ACL_FINDENTRY                                 ;
                                      7D  11 0014A  15$:  BRB     25$                                               ;
                      50          58  D0 0014C  16$:  MOVL    BUFFER, R0                                        ; 2552
                      51          57  D0 0014F        MOVL    SIZE, R1                                          ;
                      53  00000000'  EF  D0 00152        MOVL    CALL_ACMODE, R3                                   ;
                          00000000G  00  16 00159        JSB     EXE$PROBEW                                        ;
                      7C          50  E9 0015F  17$:  BLBC    R0, 27$                                           ;
                                  7E  D4 00162        CLRL    -(SP)                                             ; 2554
                      7E          57  7D 00164        MOVQ    SIZE, -(SP)                                       ;
                          00000000'  EF  9F 00167        PUSHAB  ACL_CONTEXT                                       ;
                          00000000'  EF  DD 0016D        PUSHL   ACL_QUEUE_HEAD                                    ;
              00000000G   00          05  FB 00173        CALLS   #5, ACL_FINDTYPE                                  ;
                                      7B  11 0017A        BRB     28$                                               ;
                      03          6E  E9 0017C  18$:  BLBC    (SP), 19$                                         ; 2559
                                0114  31 0017F        BRW     40$                                               ;
                      03          5B  E8 00182  19$:  BLBS    ACL_STATUS, 21$                                   ; 2561
                                0146  31 00185  20$:  BRW     45$                                               ;
                          00000000'  EF  9F 00188  21$:  PUSHAB  ACL_CONTEXT                                       ; 2562
                          00000000'  EF  DD 0018E        PUSHL   ACL_QUEUE_HEAD                                    ;
              00000000G   00          02  FB 00194        CALLS   #2, ACL_DELETEACL                                 ;
                                      5A  11 0019B  22$:  BRB     28$                                               ;
                      50          58  D0 0019D  23$:  MOVL    BUFFER, R0                                        ; 2567
                      51          57  D0 001A0        MOVL    SIZE, R1                                          ;
                      53  00000000'  EF  D0 001A3        MOVL    CALL_ACMODE, R3                                   ;
                          00000000G  00  16 001AA        JSB     EXE$PROBEW                                        ;
```

```
                                                             M  4
SYSACLSRV                                                    16-Sep-1984 01:51:51   VAX-11 Bliss-32 V4.0-742          Page 107
V04-000        ACL_DISPATCH - main ACL function dispatcher   14-Sep-1984 12:40:53   [LOADSS.SRC]SYSACLSRV.B32;1        (16)

                            59     50  E9 001B0 24$:   BLBC    R0, 30$                                      ;
                            7E     57  7D 001B3        MOVQ    SIZE, -(SP)                                  ; 2569
                     00000000'     EF  9F 001B6        PUSHAB  ACL_CONTEXT                                  ;
                     00000000'     EF  DD 001BC        PUSHL   ACL_QUEUE_HEAD                               ;
             00000000G      00     04  FB 001C2        CALLS   #4, ACL_READACL                              ;
                                   5A  11 001C9 25$:   BRB     31$                                          ;
                            50     58  D0 001CB 26$:   MOVL    BUFFER, R0                                   ; 2574
                            51     57  D0 001CE        MOVL    SIZE, R1                                     ;
                     53 00000000'  EF  D0 001D1        MOVL    CALL_ACMODE, R3                              ;
                     00000000G     00  16 001D8        JSB     EXE$PROBEW                                   ;
                            61     50  E9 001DE 27$:   BLBC    R0, 33$                                      ;
                            7E     57  7D 001E1        MOVQ    SIZE, -(SP)                                  ; 2576
                     00000000'     EF  9F 001E4        PUSHAB  ACL_CONTEXT                                  ;
                     00000000'     EF  DD 001EA        PUSHL   ACL_QUEUE_HEAD                               ;
             00000000G      00     04  FB 001F0        CALLS   #4, ACL_ACLLENGTH                            ;
                                   2C  11 001F7 28$:   BRB     31$                                          ;
                            50     58  D0 001F9 29$:   MOVL    BUFFER, R0                                   ; 2581
                            51     57  D0 001FC        MOVL    SIZE, R1                                     ;
                     53 00000000'  EF  D0 001FF        MOVL    CALL_ACMODE, R3                              ;
                     00000000G     00  16 00206        JSB     EXE$PROBEW                                   ;
                            33     50  E9 0020C 30$:   BLBC    R0, 33$                                      ;
                            7E     57  7D 0020F        MOVQ    SIZE, -(SP)                                  ; 2583
                     00000000'     EF  9F 00212        PUSHAB  ACL_CONTEXT                                  ;
                     00000000'     EF  DD 00218        PUSHL   ACL_QUEUE_HEAD                               ;
             00000000G      00     04  FB 0021E        CALLS   #4, ACL_READACE                              ;
                            5B     50  D0 00225 31$:   MOVL    R0, ACL_STATUS                               ;
                                   6F  11 00228        BRB     41$                                          ; 2510
                            04     57  D1 0022A 32$:   CMPL    SIZE, #4                                     ; 2590
                                   67  1F 0022D        BLSSU   40$                                          ;
                            50     58  D0 0022F        MOVL    BUFFER, R0                                   ; 2592
                            51     57  D0 00232        MOVL    SIZE, R1                                     ;
                     53 00000000'  EF  D0 00235        MOVL    CALL_ACMODE, R3                              ;
                     00000000G     00  16 0023C        JSB     EXE$PROBEW                                   ;
                            6C     50  E9 00242 33$:   BLBC    R0, 43$                                      ;
                            7E     03  7D 00245        MOVQ    #3, -(SP)                                    ; 2606
                                   7E  7C 00248        CLRQ    -(SP)                                        ;
                                   7E  D4 0024A        CLRL    -(SP)                                        ;
                     03 00000000'  EF  D1 0024C        CMPL    CALL_ACMODE, #3                              ;
                                   08  12 00253        BNEQ    34$                                          ;
                     00000000'     EF  DD 00255        PUSHL   PARENT_ID                                    ;
                                   02  11 0025B        BRB     35$                                          ;
                                   7E  D4 0025D 34$:   CLRL    -(SP)                                        ;
                     00000000'     EF  9F 0025F 35$:   PUSHAB  LOCK_RESNAM                                  ;
                                   1C  DD 00265        PUSHL   #28                                          ;
                               28  AE  9F 00267        PUSHAB  LOCAL_IOSB                                   ;
                            0A     5A  D1 0026A        CMPL    FUNCTION_CODE, #10                           ;
                                   04  12 0026D        BNEQ    36$                                          ;
                                   01  DD 0026F        PUSHL   #1                                           ;
                                   02  11 00271        BRB     37$                                          ;
                                   04  DD 00273 36$:   PUSHL   #4                                           ;
                                   7E  D4 00275 37$:   CLRL    -(SP)                                        ;
             00000000G      00     0B  FB 00277        CALLS   #11, SYS$ENQ                                 ;
                            59     50  D0 0027E        MOVL    R0, STATUS                                   ;
                            04     59  E9 00281        BLBC    STATUS, 38$                                  ; 2607
                            59  08 AE  3C 00284        MOVZWL  LOCAL_IOSB, STATUS                           ;
    57          00      0C  AE     04  2C 00288 38$:   MOVC5   #4, LOCAL_IOSB+4, #0, SIZE, (BUFFER)         ; 2610
                                   68     0028E                                                             ;
```

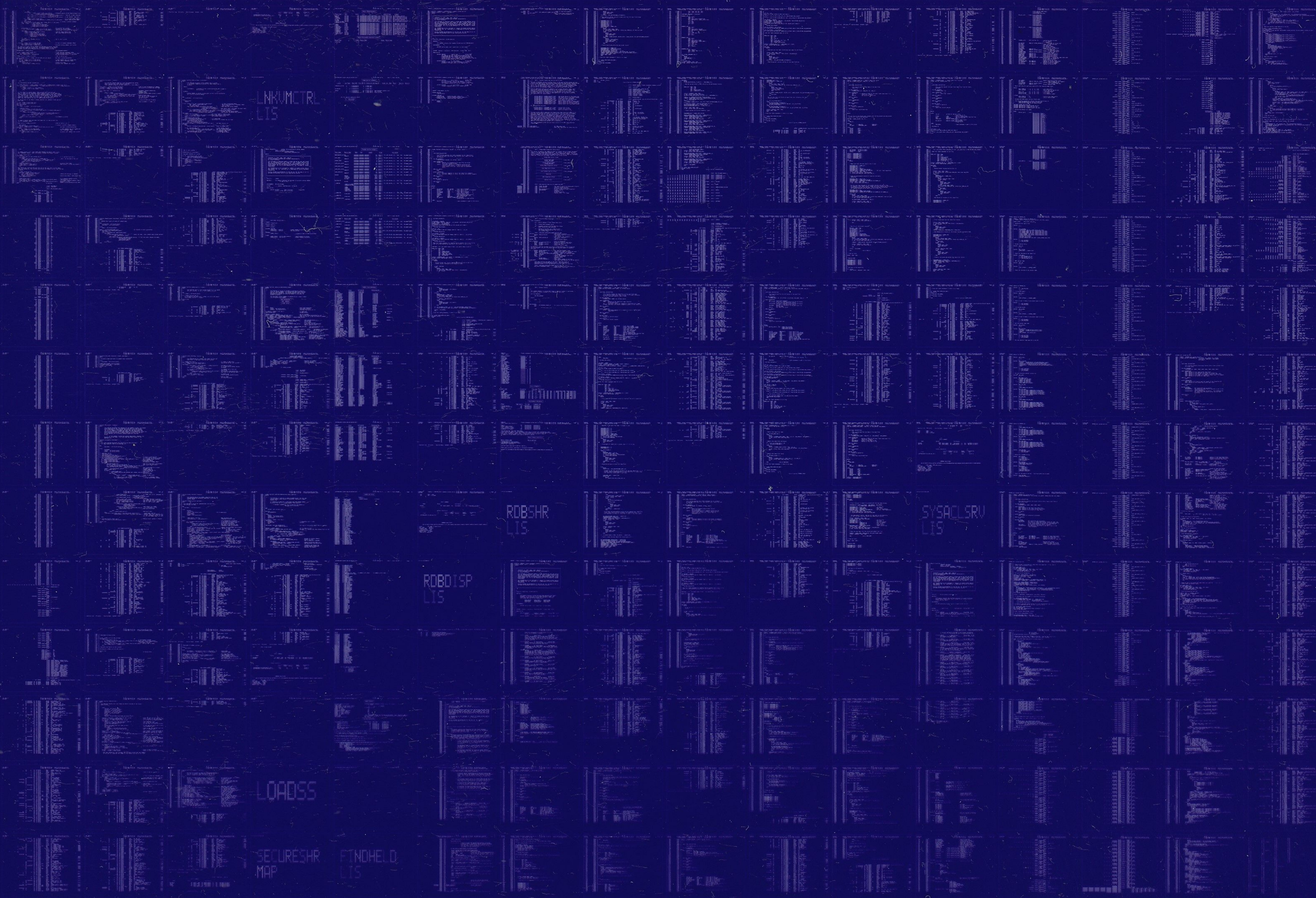
```
_$

Sy
--
AN
AS
BR
CD
CH
CI
CI
CI
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
CL
```

```
                                        3D  11 0028F        BRB     45$                                                  ; 2510
                       04               57  D1 00291 39$:   CMPL    SIZE, #4                                             ; 2617
                                        05  1E 00294        BGEQU   42$                                                  ;
                       59               14  D0 00296 40$:   MOVL    #20, STATUS                                          ; 2618
                                        33  11 00299 41$:   BRB     45$                                                  ;
                       50               58  D0 0029B 42$:   MOVL    BUFFER, R0                                           ; 2619
                       51               57  D0 0029E        MOVL    SIZE, R1                                             ;
                       53 00000000'     EF  D0 002A1        MOVL    CALL_ACMODE, R3                                      ;
                          00000000G     00  16 002A8        JSB     EXE$PROBER                                           ;
                       05               50  E8 002AE        BLBS    R0, 44$                                              ;
                       59               0C  D0 002B1 43$:   MOVL    #12, STATUS                                          ; 2620
                                        18  11 002B4        BRB     45$                                                  ;
       04        00    68               57  2C 002B6 44$:   MOVC5   SIZE, (BUFFER), #0, #4, LOCAL_LOCKID                 ; 2623
                                  04    AE     002BB                                                                     ;
                                        7E  7C 002BD        CLRQ    -(SP)                                                ; 2624
                                        7E  D4 002BF        CLRL    -(SP)                                                ;
                                  10    AE  DD 002C1        PUSHL   LOCAL_LOCKID                                         ;
                 00000000G  00          04  FB 002C4        CALLS   #4, SYS$DEQ                                          ;
                       59               50  D0 002CB        MOVL    R0, STATUS                                           ;
                       10               59  E9 002CE 45$:   BLBC    STATUS, 49$                                          ; 2635
                 02    56         04    AC  F2 002D1 46$:   AOBLSS  ITEM_COUNT, J, 47$                                   ; 2500
                                        03  11 002D6        BRB     48$                                                  ;
                                      FD65  31 002D8 47$:   BRW     3$                                                   ;
                       03               59  E9 002DB 48$:   BLBC    STATUS, 49$                                          ; 2642
                       59               5B  D0 002DE        MOVL    ACL_STATUS, STATUS                                   ;
                 50 00000000'           EF  24  C3 002E1 49$:   SUBL3   #36, ACL_QUEUE_HEAD, R0                          ; 2646
                       54 00000000G     00  D0 002E9        MOVL    CTL$GL_PCB, R4                                       ;
                          00000000G     00  16 002F0        JSB     SCH$UNLOCK                                           ;
                       50               59  D0 002F6        MOVL    STATUS, R0                                           ; 2648
                                            04 002F9        RET                                                          ; 2650

; Routine Size:  762 bytes,    Routine Base:  $CODE$ + 1A87
; 2667        2651  1 %SBTTL  'RUNDOWN_CHANGE_ACL - run down $CHANGE_ACL context'
; 2668        2652  1 GLOBAL ROUTINE RUNDOWN_CHANGE_ACL =
; 2669        2653  1
; 2670        2654  1 !++
; 2671        2655  1 !
; 2672        2656  1 ! FUNCTIONAL DESCRIPTION:
; 2673        2657  1 !
; 2674        2658  1 !       This routine is called to perform the appropriate ACL operations.
; 2675        2659  1 !       The code is checked for validity and, when necessary, the buffer
; 2676        2660  1 !       is probed for the desired access.
; 2677        2661  1 !
; 2678        2662  1 ! CALLING SEQUENCE:
; 2679        2663  1 !       RUNDOWN_CHANGE_ACL ()
; 2680        2664  1 !
; 2681        2665  1 ! INPUT PARAMETERS:
; 2682        2666  1 !       NONE
; 2683        2667  1 !
; 2684        2668  1 ! IMPLICIT INPUTS:
; 2685        2669  1 !       PARENT_ID: lock ID of parent for ACL locks
; 2686        2670  1 !
; 2687        2671  1 ! OUTPUT PARAMETERS:
; 2688        2672  1 !       NONE
; 2689        2673  1 !
; 2690        2674  1 ! IMPLICIT OUTPUTS:
; 2691        2675  1 !       NONE
; 2692        2676  1 !
; 2693        2677  1 ! ROUTINE VALUE:
; 2694        2678  1 !       1
; 2695        2679  1 !
; 2696        2680  1 ! SIDE EFFECTS:
; 2697        2681  1 !       All ACL locks taken out by user mode $CHANGE_ACL calls, plus the
; 2698        2682  1 !       parent lock, are dequeued.
; 2699        2683  1 !
; 2700        2684  1 !--
; 2701        2685  1
; 2702        2686  2 BEGIN
; 2703        2687  2
; 2704        2688  2 IF .PARENT_ID NEQ 0
; 2705        2689  2 THEN
; 2706        2690  3     BEGIN
; 2707      P 2691  3     $DEQ (LKID  = .PARENT_ID,
; 2708        2692  3          FLAGS = LCK$M_DEQALL);
; 2709        2693  3     $DEQ (LKID  = .PARENT_ID);
; 2710        2694  3     PARENT_ID = 0;
; 2711        2695  2     END;
; 2712        2696  2
; 2713        2697  2 1
; 2714        2698  1 END;                                    ! End of routine RUNDOWN_CHANGE_ACL


                                   000C 00000          .ENTRY  RUNDOWN_CHANGE_ACL, Save R2,R3              ; 2652
                  53 00000000G  00  9E 00002          MOVAB   SYS$DEQ, R3                                 ;
                  52 00000000'  EF  9E 00009          MOVAB   PARENT_ID, R2                               ;
                  50            62  D0 00010          MOVL    PARENT_ID, R0                               ; 2688
```

```
                                          14  13 00013            BEQL    1$                                    ;
                                          01  DD 00015            PUSHL   #1                                    ; 2692
                                          7E  7C 00017            CLRQ    -(SP)                                 ;
                                          50  DD 00019            PUSHL   R0                                    ;
                              63          04  FB 0001B            CALLS   #4, SYS$DEQ                           ;
                                          7E  7C 0001E            CLRQ    -(SP)                                 ; 2693
                                          7E  D4 00020            CLRL    -(SP)                                 ;
                                          62  DD 00022            PUSHL   PARENT_ID                             ;
                              63          04  FB 00024            CALLS   #4, SYS$DEQ                           ;
                                          62  D4 00027            CLRL    PARENT_ID                             ; 2694
                              50          01  D0 00029 1$:        MOVL    #1, R0                                ; 2698
                                              04 0002C            RET                                           ;

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 1D81


; 2715          2699  1
; 2716          2700  1 END
; 2717          2701  0 ELUDOM





;                                 PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;     $OWN$                      1171  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;     $PLIT$                     1396  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;     _LIB$KEY0$                   42  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;     _LIB$STATE$                 664  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;     _LIB$KEY1$                  213  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;     $CODE$                     7598  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;     .  ABS  .                     0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)




;                            Library Statistics
;
;                                            -------- Symbols --------       Pages      Processing
;       File                                 Total    Loaded   Percent       Mapped     Time
;
;     _$255$DUA28:[SYSLIB]LIB.L32;1          18619       211         1       1000        00:01.8
;     _$255$DUA28:[SYSLIB]TPAMAC.L32;1          42        28        66         14        00:00.2



; Information:   1
; Warnings:      0
; Errors:        0
;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SYSACLSRV/OBJ=OBJ$:SYSACLSRV MSRC$:SYSACLSRV/UPDATE=(ENH$:SYSACLSRV)

; Size:          7598 code + 3486 data bytes
; Run Time:         03:19.4
; Elapsed Time:     06:02.3
; Lines/CPU Min:        812
; Lexemes/CPU-Min: 37990
; Memory Used:  1462 pages
; Compilation Complete
```

0220 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0221 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY